Tool planning in batch manufacturing.

PATANGE, Vinay B.

Published version

PATANGE, Vinay B. (1993). Tool planning in batch manufacturing. Doctoral, Sheffield Hallam University (United Kingdom)..

TOOL PLANNING IN BATCH MANUFACTURING

Vinay Baburao Patange

A thesis submitted in partial fulfilment of the requirements of Sheffield Hallam University for the Degree of Doctor of Philosophy

SHEFFIELD HALLAM UNIVERSITY
SHEFFIELD
NOVEMBER 1993

ABSTRACT

TOOL PLANNING IN BATCH MANUFACTURING

by
Vinay Patange

This work concentrates on the newly growing science of managing tooling in conventional manufacturing. Various Tool Management (TM) problems and the approaches suggested by other researchers to solve these problems are given. This work establishes the basic structure of TM applicable to a conventional manufacturing. Systems Analysis and Design Methodology (SSADM) is used to study the information needs of a typical TM System.

It is stressed that the majority of TM problems are due to unavailability of correct information. Success of any TM system depends upon having a good Information System. This work focuses on the tool planning problems in batch manufacturing. The causes of tool planning problems are discussed. The research aims to develop a generic methodology for planning the tools. The information required to carry out the primary functions of any Tool Planning System (TPS) is identified. The fundamental characteristics of different tools from the planning perspective are studied in detail. The principles on which a generic TPS could be designed are laid out. The mechanism of a Tool Planning System is developed.

The TPS model is implemented using Foxpro, a DBMS. This model illustrates the concepts of planning tools with the information that can be obtained from other functions of manufacturing. The effectiveness of the developed TPS model is investigated using simulation. The impact of the TPS on the performance of a typical Job Shop Environment is studied and compared with other models with traditional stock control approaches.

A suitable statistical method is used for analysing and comparing the simulation results. The advantages and the limitations of the TPS are discussed. Some of the potential benefits include, very low tool shortages, minimum number of purchase requirements and better estimation of tool inventory levels. Furthermore, the TPS acts as a firm guideline for planning the tools in time buckets.

PREFACE

This thesis is submitted to the School of Engineering of Sheffield Hallam University for the degree of Doctor of Philosophy. This research was carried out in the Division of Design and Manufacture of School of Engineering.

I would like to express my deepest gratitude and appreciation to my supervisor **Dr.D.T.S. Perera** for his guidance and encouragement throughout the course of this work. I would like to thank my research colleagues, the technical, administrative and lecturing staff within the School of Engineering for their help and support.

The results obtained and the theories developed during the course of this research are to the best of my knowledge original, except where the reference is made to the work of others.

V.B. Patange

November 1993

CONTENTS

CONTENTS

GENERAL OVERVIEW

Tooling forms an important part of the foundation on which the new systems like Computer Integrated Manufacture (CIM) and Just In Time (JIT) are built. Many manufacturing professionals have realised that the success of these systems is closely tied to tooling.

The definition of tooling as given by Broom [1967] is,

"all equipment and special fixtures that the system can draw on and use during the set up and operation of a machine or an assembly process."

Melnyk [1988] groups the various items that satisfiy the above definition into three major categories, viz; Transportation tooling, Set up tooling and Production tooling. This indicates that Tool Management embraces many facets of manufacturing and therefore becomes an integral part of the business activity. Meeting delivery schedules is crucial to Production Managers and cannot be achieved unless the necessary resources are planned and managed properly. Control of tools is just as important as the management of other resources like people, material and machines. Reports indicate that 16% of scheduled production is delayed due to tooling shortages [Mason, 1988]. Tooling is a key manufacturing resource which has been overlooked in the past. Therefore, better understanding of tooling and its features is required among manufacturing professionals and researchers.

Any Tool Management System strives to ensure the availability of tools at the required time, at the required place, in the correct quantity and condition [Mason, 1988]. According to Chapman [1990], Tool Management is defined differently by different people. Tool Management is not simply the control of tool inventory, but encompasses many diverse activities [Chapman, 1990]. These activities include Tool Transportation, Quick Change tooling systems, Tool Identification, Tool Presetting, Post processs gauaging systems and Electronic tool ordering system. Tyner [1988] addresses Tool Management as the combination of problems related to flow of tools, tool presetting and tool crib. His study included the Ergonomics (Human Factors) for safety and emphasis on Employee Education and Training for waste elimination from the tool management perspective. It is important that such activities are coordinated and synchronised for achieving the objectives set out for a Tool Management System. Therefore, it seems that it is necessary to understand the various elements of a typical Tool Management System and establish their inter-relationships.

Traditionally, the management of tools was left to the machine operator or storeman. The decisions on the issue of the new tools and disposal of the old ones were taken by the operators. There were no methods of predicting the demand of different types of tools. This resulted in higher tooling inventory and therefore higher manufacturing costs. About 50% of perishable tools (tools that wear out and are then disposed) in American industries have been reported to be obsolete [Mason, 1988]. Many authors have emphasised the problems of controlling tool inventory because this added significantly to the cost of the product. Some industries adopted the practice of having a central tool store as against the distributed stores. This gave an opportunity to control the tools centrally and thereby avoiding the tool proliferation. In many organisations, each operator had an individual tool box, which increased total volume of tools outside the tool stores. As a result, the expenditure on these tools increased. There were no means of controlling this cost.

Lately, industries have realised the importance of this science. There are computerised tool management systems available in the market, which can be implemented directly, thereby replacing the manual tool transactions in the tool stores. However, some industries have not been able to take the full potential of the facilities provided by these computerised systems [Mason, 1991]. This has been mainly due to the incorrect use of implementation methods. This research work encompasses the detailed study of a typical tool management system and its primary functions and extends the work on tool planning in batch manufacturing.

2

LITERATURE SURVEY

2.1 INTRODUCTION

With the advent of advanced manufacturing technologies like Flexible Manufacturing System (FMS), CAD/CAM and CIM, new management practices such as MRP and JIT were adopted by the industries. The new concepts such as FMS, in particular, faced constraints in its operational flexibility due to availability of both cutting and non-cutting tools. There were other factors like tool magazine capacity in FMS which greatly reduced its flexibility in part variety [Perera, 1988]. These problems instigated the manufacturing professionals to develop better methods of Tool Management in FMS. Thus, a new science of managing tooling was evolved.

A literature survey was carried out to understand the current trends of Tool Management in manufacturing industry and problems associated with it. The science of Tool Management (TM) is still in its early developments and many industries are not yet aware of this potential cost saving area [Mason, 1988]. The first international conference on TM [Michigan, 1988] shows that individual researchers identified problems relevant to their type of business for e.g. the Airline Industry reported problems in scheduling their critical tools for aircraft maintenance at different destinations worldwide [Lidbury, 1989]. They proposed solutions applicable to their type of operations only (the solutions to the airline TM problems are suugested by having a good bar coding system for tools). Similiarly, within the metal cutting industries, different industries pay attention to the type of tooling they use most commonly. Manufacturing processes involving large proportions of finishing (e.g. buffing operations) require higher quantity of perishable tools, whereas, the conventional metal cutting operations and assembly activities require higher number of returnable (or durable) type of tools. Therefore, it is important to study the features of different tools, on which management procedures could be developed. There is a need to identify primary functions of TM with a unified approach. A generic structure of TM can be constructed which would consist of these functional elements. This could form a basis for manufacturing professionals to develop new techniques to solve their TM problems.

Some of the causes identified by Carrie and Bititci [1988] for the failure of a typical tool management systems include complexity of TM logistics, lack of shopfloor discipline, lack of correct and complete information to management, lack of computer based planning systems and incompatibility of production machines with control equipment. Lack of shopfloor discipline led to the conflicts in coordinating machine shop management with the tool stores.

The characteristics of FMS are different from conventional manufacturing. Therefore, the TM problems are also different in these manufacturing environments.

Consequently, the literature on TM had to be classified into two major categories, viz; TM problems specific to the FMSs and TM in conventional manufacturing systems. The following sections describe various TM problems in FMS and non-FMS and discuss how they are different from one another.

2.2 TM RESEARCH ON FMS

The FMS allowed greater flexibility in their operations. However, it seems that in most systems tooling constraints flexibility. These FMS constraints attracted many researchers [Ber 1982, Falkenberg 1986, Sackett 1984, Hutchinson 1982] to develop techniques and establish discipline for management of tools in FMS. The research on FMS tool management increased dramatically during the 1980s. As a result, the savings on tooling in FMS encouraged researchers to develop better methods and sophisticated techniques (such as on-line tool information system) to solve the TM problems. Thus tool management became a popular research topic among many manufacturing professionals.

Kuchinic and Seidmann [1988] grouped all the tool management issues into three categories, viz; tool specific issues, machine level issues and factory level issues. The tool specific issues included problems like tool life, tool economics, tool standardization and Spares Management. The Machine level issues involved tool magazine capacity constraints, automatic tool changer capabilities (ATC) and tool wear/ breakage monitoring systems. The factory management level encompasses the problems of tool allocations to various machines, tool requirements planning, tool inventory and tool procurement. Several researchers [Stecke and Solberg 1981, Stecke 1983, Rajagopalan 1986, O'Grady and Menon 1987] noted that short to medium term FMS production planning must consider the limitations imposed by the tool magazine capacity. The limited tool magazine capacity implies that proper tool allocation and scheduling procedures must also be used in order to achieve the performance potential of the FMS. Bao [1980], Ramlingam [1976] and Kendall [1976] addressed the problem of tool replacement schedules in FMS. They used dynamic programming techniques to optimise these schedules. These problems indicate the complexity of tool planning decisions in FMS.

Amoako-Gyampah et al [1992] highlights the problem of tool allocation and tool scheduling for the FMSs. They suggested three tool allocation strategies viz; bulk exchange, tool migration and resident tooling. One of the important production planning problems identified by Stecke [1983] was related to the allocation of operations and required tooling for the selected parts among the machine groups in FMS. Ramalingam and Balsubramaniam [1976] developed mathematical models to solve machine loading and tool allocation problems in FMS.

Carrie and Perera [1986] observed that the part type selection problem in FMS is one of the production planning problems severely constrained by tooling. Chakravarty and Shtub [1984] used a Group Technology approach to solve the problems of part type selection. They identified parts having similar machining requirements and thereby identifying similar tooling. The capability of having alternative rerouting options for parts make production planning in FMSs much more difficult than traditional production lines and job shops. Ventura [1990] explained how part grouping would ease FMS production planning problems.

Rhodes [1986] found that FMS production parameters like the number of parts and batch operation times influenced the tool management variables like number of tools, tool magazine capacity and tool exchanges. These parameters distinctly characterise FMS tool management requirements.

Ranky [1986] stressed the need for an integrated tool database for FMS real time control. Tool management involves a variety of activities which makes it essential to have on-line information for the dynamic environment. His study showed that each FMS needs tool management tailored to its requirements.

2.3 USE OF FMS TOOL PLANNING TECHNIQUES FOR CONVENTIONAL MANUFACTURING

Researchers have developed few methods of planning tools in FMSs. Perera [1991] highlighted one of the major differences between conventional manufacturing and FMS. Formal planning systems steer the events in FMSs. A detailed preplanning system is essential to ensure the uninterrupted flow of parts within the FMS. It is also important that real time data is available to overcome the tool planning problems in FMS. Perera's [1988] tool flow simulator provides some visibility into expected tooling shortages within FMS environment.

The tooling problems like tool flow, tool exchanges, tool magazine capacity etc. are usually not found in conventional manufacturing, unless it is a very highly automated production system with facilities like automatic tool transportation, automatic tool changing systems or automatic part loading. Tooling information is stored centrally in the FMS executive computer. There is no real time data available through such central computer in conventional manufacturing. Therefore, many of the TM techniques such as tool allocation strategies and tool flow using simulation do not apply to traditional systems. These features of manufacturing are found only in FMS. Hence, it is not feasible to use the FMS TM techniques for conventional manufacturing.

So far, the savings on tools in FMS were important in a way that FMSs are expensive to run and tools constrained the flexibility. Therefore, most researchers diverted their attention to FMSs. As a result, there are very sophisticated techniques now available for FMSs, but the TM in traditional systems is still in its embryonic stage. Therefore, there is a need to have better TM techniques for conventional systems.

2.4 TM RESEARCH FOR CONVENTIONAL MANUFACTURING

The first evidence of work found on TM in conventional systems consisted of the mechanism of a generic tool control system. Galligan and Mokris [1981] modelled the mechanics of a generic structure of TM Information System. They identified functional requirements of an integrated tool control system and grouped into three major categories, viz; Tool Inventory Control, Engineering Change Control, Tool Consumption and Cost Control. However, it appears that the planning aspects of tools were not developed at that stage. These facilities were designed and incorporated on a computerised system. They also developed the tool data system which would provide information such as tool master, tool purchase, tool transactions and tool consumption. They suggested the option of designing the central tool database from where the appropriate information could be provided to the required places. It is now widely accepted that causes of many TM problems are centred towards database management systems [Galligan et al, 1981]. The detailed discussion on TM database management systems is available in Chapter 5.

A typical TM in conventional batch manufacturing system may involve all kinds of activities from the stage of tool purchase to tool maintenance and tool disposal. Long[2] [1991] lists some of the TM activities, which are as follows.

Tool Engineering	Tool Design
Tool Procurement	Tool Rework
Tool Inspection	Tool Storage
Tool Presetting	Tool Accountability
Tool Planning	Tool Inventory Control

Ideally, every TM system should be capable of performing the above mentioned functions. Due to the range of activities involved in any TM system, it is difficult to co-ordinate them and in many cases, it is usually beyond the capacity of human control. Additionally, the variety of tooling makes it even more difficult to manage them, because different types of tool need different management procedures. It is, therefore, necessary to define what primary functions are required by any formal TM system and how these functions relate to other functions of manufacturing. The survey indicated that no researcher in the past has made any attempt to establish the exact structure of TM applicable to conventional manufacturing system. Due to the complexity of TM activities, a systematic approach is required to study each component of TM in detail.

The survey also showed that within Tool Engineering Control, there is evidence of work on Tool selection procedures [Maropoulos[1] et al, Zhou 1988, Traughber 1986, Weimer 1983]. These techniques are linked with the CAD/CAM to save the costs incurred in inappropriate selection of tools.

Melnyk[1] et al modelled the simulation of a machine shop with machines and tools. The work was focused on the impact of an alternative tooling assignment rules on the operation of the shop, with varying levels of tool availability and job priority rules. This study was related specifically to tool assignment rules.

Besides the above work in conventional manufacturing, only Wassweiler [1982] gave a new approach called Tool Requirements Planning (TRP). This method determines the tooling requirements from the process plans and schedules them in accordance with production schedules. The technique can schedule the critical tools on hourly basis. The tool allocation schedule is very precise, however, this system is limited to heavy fabrication shops of the make to order type with very high product variety and low volumes. It is most suitable in heavy fabrication type environment, where special tools (usually more expensive than standard tools) form a limited resource and where sharing of such tools is more frequent.

The need for an alternative approach to planning tools for batch manufacturing arises due to the four basic differences in the features of these two production environments, which are laid out in the following table. This has significant influence on the planning procedures.

System Features	Heavy Fabrication	Batch Manufacturing
Product Variety	Very high	Medium
Production batch size	Very small, usually one	Large
Planning Horizon	Large (months/years)	Short (days/weeks)
Type of tools used	High usage of multiple and complex tooling. Requires tool assembly and kitting.	High usage of single tools.

In Wassweiler's Tool Requirement Planning (TRP) method, the tools are scheduled in line with the part schedule. The tools are allocated to a single job and the manufacturing lead time of such jobs in heavy fabrication is usually longer than the jobs in batch manufacturing. Allocation of tools to individual jobs is not feasible in batch manufacturing because the production lead time of components is small. Hence, an alternative approach of allocating tools to the batches rather than the individual components is required.

Usually, there is a higher usage of simpler and single tools in high volume batches, which requires a different method than that for complex multiple tools. The TRP accounts for tools requiring assembly, which uses the concept of a Bill of Tools (BOT) similar to a Bill of Material (BOM). The BOT concept is used for scheduling the tool kitting activities.

Since the batch sizes are usually larger in batch manufacturing, the demand of certain tools is on a regular basis as against the unpredictable demand in heavy fabrication. This feature will have a significant impact on the tool planning strategies and the associated tool planning methods. Therefore, the TRP method can not be used in batch manufacturing.

There are MRP type methods available for planning and allocation of material. It is necessary to understand why such techniques cannot be used for tool planning. Melnyk [1989] has identified the fundamental characteristics of tools. It is like material which can not be used at more than one place. Secondly, when the tool is returned to the stores, there is no change in its stock quantity, but its total tool life has decreased. In other words, the process capability of tool has decreased.

Therefore, the tool planning technique should consider both the above characteristics. A hybrid approach of classical stock control together with the logistics that would account for the tool life can be used to plan the tools. In order to develop such a tool planning system, information from various external functions such as Process Planning and Production Planning is required to carry out the necessary data processing.

It can be summarised that there is a lack of work on tool planning techniques, in particular, which would be applicable in batch manufacturing environment. However, there are commercial TM software packages available, which perform basic TM like issue, return and stock holding. The discussion on the TM software is given in the following section.

2.5 DEVELOPMENTS IN TM SOFTWARE

Allock [1986] reports that Siemens developed a software for a Flexible Manufacturing Cell (FMC) that comprised a module for tool requirements. For every new job, before loading, the gross requirements including the remaining tool life are identified to ensure the completion of jobs. It is claimed that the system is on-line, as it computes the net requirements for the next job immediately after the completion of the first job. However, this article does not clearly explain exactly how the tool requirements are identified. Additionally, it lacks explanation on tool life estimation for individual tools on the part type basis. Usually, in an FMS environment, the machining content of jobs will vary significantly from one part to another. Unless very sophisticated methods for predicting tool life consumption are used, it is very difficult to ensure the job's completion.

Electronic Data System (EDS) (a Software House) is assisting General Motors in building computerized tool management systems. The other companies who have installed such a system includes Kennametal, GTE Valenite and Sandvik [Mason, May 1991]. Most of these systems offer facilities like tool specification, purchasing, presetting, identification, inventory control and tracking. Mason [May 1991] lists several software houses supplying dedicated tool inventory and tracking software. Some of these software include, Cribware, TMS-2000, ATICTS, ITC's Toolware, Tool Location Control and Microtoolware, Kavon's Cribmaster, Spacesaver's TCS, Sycon's PC-Toolcrib.

Our survey of software houses supplying commercial computerised tool management system included Amazon's CTMS and Cincinnatti Milacron's TMS-2000 and Tooltrak. TMS-2000 offers facilities like Inventory Control, Bill of Tools, Purchasing information (details of tool suppliers) and Gauge Control. The additional module Tooltrak, provides facility to locate the tools, assign the critical tools to the workcenter or the product and keep the track of individual tool usage or rate of consumption. It also allows you to set the tool reorder level to maintain the required inventory levels.

Amazon's CTMS provide the facilities for tool transactions (issue and return), allows you to build the tool kits and generate the Bill of Tool (BOT) structure with kits and assemblies, generates the Purchase Requirements based on the current inventory levels and the reorder levels. It also offers facility for recording the details related to tool presetting, tool calibration and tool inspection.

It appears that the above software offer facilities that are important to achieve tool stores and to some extent tool stock control requirements. However, none of them provide a facility to plan and schedule tools based on the master schedule. It is realised that there is a need to bridge a gap between Production Planning and Tool Management through a Tool Planning function.

2.6 PROBLEM DEFINITION

The aim of the research programme is *to develop a tool planning system for batch manufacturing.*

There are two major problems in batch manufacturing that are associated with tooling. One is the excessive tool inventory in tool stores and the other is the production stoppages due to unavailability of tools. These problems indicate that tools are either not used to the fullest extent or not planned properly. Although, tool stores records show that they are available, it is not known where they are located and in what condition. This often leads to expediting tool purchases and thus increasing the inventory. Therefore, methods for planning the tools on the basis of Production Plans need to be developed. It is essential for any tool management system to have a mechanism to plan tools in order to ensure their availability. This work explores such a possibility by developing a generic Tool Planning Methodology in conventional manufacturing. Moreover, this appeared to be the potential research area where significant original work can be produced.

3

FUNDAMENTALS OF TOOL PLANNING

3.1 INTRODUCTION

The manufacturing industries of the 80s used Material Requirement Planning (MRP) type planning system in the batch manufacturing environment. The factory schedules were created based on forecasted demand, backlogs or safety stock levels, interplant orders and customer orders. The resources such as men, machines and material were taken into account by production planning but tooling was neglected.

The availability of tools is of paramount importance for the smooth running of production according to production schedules. It is, therefore, necessary to include tooling as an additional resource in the process of production planning. This means that, while developing an MPS, if the tools are accounted for, then the formation of the MPS will be influenced by the tooling. Hence, it is essential to study the effect of new TM techniques on the MPS which can be achieved by integrating MRP-II with the TM. Tool Planning froms a prominent element of any TM system.

Having understood the importance of tooling in manufacturing control, it is then necessary to see how such a resource can be linked/integrated with other production planning and control functions. For example, is it possible to integrate TM with

Manufacturing Resource Planning (MRP-II) ? Can the MPS be created on the basis of tooling availability ? or the tools to be planned according to the MPS ? Figure 3.1 shows the complexity of tool planning decisions in the context of MRP-II. Fig 3.2 shows the role of Tool Planning in MRP-II environment. This chapter will discuss such a possibility of bridging the production planning function with the TM and highlight the advantages by doing so. An attempt has been made here to construct the generic structure of a typical TM system with its primary functions.

3.2 SELECTING A SUITABLE TECHNIQUE FOR ANALYSING THE TM SYSTEM

In order to develop methods for planning tools, it is necessary to study the various functions of a TM system, their inter-relationship and their relationship with other functions of manufacturing such as CAD, Production Planning and Process Planning. There are methods available for analysis of complex systems, such as ICAM definition methods (IDEF0 and IDEF1), Structured Design Method (SDM) which is principally based on Jackson Structured Programming (JSP) and Structured Systems Analysis and Design Methodology (SSADM) which is derived from SDM.

SSADM is a well established methodology approved by the U.K. government. It focuses on the analysis of business requirements for, and the design and specification of, an application database and software [CCTA, 1990]. SSADM has now been adopted by many organisations and has become the leading system analysis and design method in the U.K. AUTO-MATE is a Computer Aided Software Engineering (CASE) tool, which has now been widely used for systems analysis and design. It was originally developed by Learmonth and Burchett Management Systems (LBMS) in the U.K [LBMS]. SSADM in conjunction with AUTO-MATE was an ideal choice and hence, it was selected for this study.

In this work, primarily two analysis techniques from SSADM are being used, which are the Functional Analysis and Data Analysis. The functional analysis gives a thorough understanding of the various functions of TM system and its relationship with the other manufacturing functions. The data analysis gives the database specifications for the required system (in our case it is the TM system) in a normalised form (Data Analysis and normalisation is explained in chapter 4).

3.3 FUNCTIONAL ANALYSIS

Functions or activities need certain information to achieve their objectives. The analysis is carried out top down, level by level. High level functions being identified first with lower levels being introduced by successive functional decomposition [Cutts, 1987].

The functional analysis produces what is known as the 'Data Flow Diagram (DFD)'. The DFDs contain the functions, sources of data flow, destinations of data flow and data stores. Figure 3.3 shows the diagram conventions used in DFDs. The DFDs represent the user's view and therefore fully understandable by the user. Hence, there are no fixed rules governing the number of functions that should be shown on a single DFD.

From the higher levels or level 1, the functional decomposition provides more detailed information by zooming in on any or all of these rectangles. Some functions may have more levels than others. The picture of such a decomposition is shown in Fig 3.4. The meaning of various objects used in creating DFDs are explained below.

(1) Functions :-

Functions are represented by rectangles and form the dominant feature of DFD. Each function is given an identification number, a single number at level 1 and compound numbers at subsequent lower levels. The respective authority responsible for this function is associated with this identification number. A short description of actual activity is given in the rectangle.

(2) Sources and destinations for data :-

Ellipses represent external functions. Each ellipse may be used at more than one place in a single DFD. In order to show that there is more than one ellipse representing the same external function, a line can be inserted into the top left hand corner of the ellipse. This line is then present in those ellipses which reoccur in the diagram.

(3) Data flow :-

Data flows are represented by arrows. Each arrow represent the flow of data element with its unique name. The single headed arrows means flow of data in one direction, whereas, double headed means data flow in both directions. The physical movement of tools is not shown on DFD, however, it is important to understand how the data related to tool movement is generated and flows through the system.

(4) Data stores :-

These are represented by open ended rectangles. They represent files, private reference books or any form of data store within the system. Each data store has a name and is associated with its unique identification. A double bar at the closed end of the rectangle indicates that this data store is repeated elsewhere on the DFD.

With the help of the above conventions all the required DFDs were created. The following section gives the overview of the TM system from the information system point of view (from Figs 3.6 to 3.9).

3.4 INTERACTION OF TOOL MANAGEMENT WITH OTHER FUNCTIONS OF MANUFACTURING

The process of analysis begins by creating the 'Context Diagram' (at level '0'). This DFD specifies the scope of the system (which is TM, in this case) by defining its boundary within the focused environment. In this case, the boundary of TM system is identified within the conventional/batch manufacturing environment (Fig 3.6).

The major manufacturing functions interacting with the TM system are Production Planning, Process Planning, Purchase Department, Computer Aided Design - Computer Aided Manufacture (CAD/CAM), Shop Floor Control and Tool Supplier. All these are treated as external functions.

Functions like Purchase and Tool Supplies may or may not be included in the TM system depending upon the working practices of the organisation. For example, Tool Requisition function can be treated either independently or as a part of other General Purchases. Similarly, tool suppliers are treated external to the TM system of the organisation, because each supplier has his own tool catalogue and in practice, the TM system may buy different tools from different suppliers.

The external entities are represented by ellipses and the focused system is shown by a rectangle. Figure 3.6 shows the proposed flow of data elements that any ideal TM system would possess. The flow of information as represented in Fig 3.6, between the TM and other functions would be valid for any ideal TM. The relationship between the six external functions (as identified earlier) and TM is discussed below.

Production Planning informs TM about the proposed MPS together with the aggregate production plans, if any, and receives the report about the feasibility of such an MPS from the viewpoint of tooling availability.

Shop Floor Control (SFC) has close interaction with TM, where SFC keeps the tool consumption records (or historical records) on either the basis of workcenter or the production order, as the case may be.

CAD can play a prominent role in TM. The new and existing products can be designed in such a way that the existing tooling can be used to fabricate them. This is also known as Tool Variety Reduction in TM terms.

Ideally, CAD should have an access to Tool Master Database to know the already existing tools or most commonly used tools for certain operations. Additionally, all the new tools introduced by either Tool Engineering Change Control (a function of TM) or newly required, as specified by CAD (e.g. form tools for special geometries) need to be included in the Tool Master Database.

Such measures for tool variety reduction have already been undertaken by an American multinational. Fig 3.5 shows a range of product variety (Door Handles) with identical geometric features. Any changes in the front end shape would require a new type of tool (in this case, different form tools will be required by different components from the high volume production point of view). Small modifications in the geometry of the front end have resulted in tool variety reduction by more than 50% for that family of components, without sacrificing their functional aspects.

Tool Procurement can be conducted by sending the appropriate tool requirements to the Purchase Dept. In return, the Purchase Dept. can inform the expected date of tool receipts and delivery details to the TM, which would assist TM in planning the tools. Similarly, the tool purchase cost database can be maintained by Tool Purchase Dept., which would be useful for both tool costing and tool budgeting purposes.

The information such as Tool Master Database (which would ideally contain information on tool specification) is essential to process planners to make the machining processes (or fabrication, as the case may be) more effective by selecting appropriate tools. In return, the TM can have an access to information about the part details (like part number) for which the tools are being selected. This would facilitate the TM to establish the relationship between the part types and associated tool consumption.

Usually, a tool purchase engineer (also termed as 'Tool Liaison Engineer') develops and maintain relationships with the tool suppliers by obtaining the information on Tool Catalogues, Tool Engineering Specification and Delivery Service Levels. Some of the most important functions of tool management are listed in the following section.

3.5 FUNCTIONS OF TM IN CONVENTIONAL MANUFACTURING

Having drawn the boundary of the TM system (in Fig 3.6), it is then decomposed into various functions of TM at the next level which is level '1' (Fig 3.7). All the external entities identified at level '0' and the related data flow is retained at this level.

The functions of TM are also represented by rectangles. The data flow from or to the external entities is connected to the newly created functions. The additional flow of data between the functions and the data stores was drawn and the final DFD was produced (Fig 3.7).

All the activities related to TM were grouped into five major functions and each function was assumed to have certain responsibilities. These are listed below,

(1) Tool Store Services

(2) Tool Planning

(3) Tool Requisition

(4) Tool Engineering Change Control

(5) Tool Inventory Control

Major functions listed above at level '1' which were then decomposed into sub-functions at level '2' (Fig 3.8 shows the Tool Store Services and Fig 3.9 shows the Tool Planning). Other functions were not decomposed as the focus of the research was on Tool Planning. The details of the activities involved to achieve the objectives of these functions are stated in the following sections.

(1) <u>Tool Store Services</u> :-

The basic activities that any tool store would be responsible for are,

(a) Issue, return, storage and transport of tools

(b) Keeping the record of location and the condition of tools.

(c) Inspection of new and used tools before storage.

(d) Updating the stock details.

(e) Generate tool consumption information (for consumable) from tool usage data received from shop floor and update tool history records (in case of returnable).

(f) Sending the requisition for the purchase of "C" class tools.

(g) Follow up of tool preparation schedule.

(2) <u>Tool Planning</u> :-

Tool Planning is anticipated to have the following sub-functions.

(a) Generate Tool Requirement Reports.

(b) Send the requisition for the purchase of critical tooling.

(c) Check the feasibility of production schedules from the view point of tool availability and informing the Production Planning Department about the tooling shortages.

(f) Schedule the critical tools according to the job order, and plan the tool preparation activities. Generate any necessary tool preparation schedule and send it to the Tool Store Services.

(3) Tool Requisition :-

Traditionally, tool purchase activities fell under General Purchase Department. However, in this study, they are considered as part of tool management as this assisted in studying the relationship between tool requisition and other functions of tool management. The functions of this department are,

(a) Generate Purchase Orders for tools on request from tool requisition list.

(b) Keep the updated record of tools ordered, expected date of receipts and provide to the tool planning with the tool awaiting list.

(c) Compile the tooling costs and maintain tool cost database to be used by tool inventory control.

(d) Evaluating the tool suppliers for their service.

(e) Liaising with the tool suppliers and develop relationship with the suppliers.

(4) Tool Engineering Change Control :-

This department will be mainly responsible for controlling the changes in the design of tools which may occur either due to the product design change or the changes in the efficiency of the machining processes (or process capability). When such changes take place, better control is required for the successful introduction of new tools. The main objectives of this function are stated below.

(a) Decide the application area of the tools.

(b) Coding and classification of new tools on the basis of their engineering specifications.

(c) Define the tool structure and enter the tool kit details into the Bill of Tool (BOT) database.

(d) To estimate the total tool life on the basis of their engineering specification.

(e) To specify whether the tool should be purchased or fabricated in-house.

(5) Tool Inventory Control :-

There is always a trade off between the investment in maintaining the tool inventory levels and the cost of administering the manufacturing resources to run the production without any tooling shortages. The important function of this department is to develop the tool inventory control strategies. Other objectives of this department are,

(a) To develop a mechanism to make the decisions regarding:

- Lot size of orders
- Setting appropriate reorder and safety stock levels (This factor is important to maintain the stock levels of perishable and semi-perishable tools.)
- Procurement lead time of tools

(b) Use tool store capacity information to develop inventory control strategies.

(c) To carry out detailed A-B-C analysis of tools from the tool cost database.

The earlier section gave a general overview of the TM functions and their relationships. The successful operation of any TM system depends on how well the information is made available to these functions and the logistics of their data processing to make the desired TM decisions. This shows the complexity of decision making process in TM.

It was anticipated that the benefits of researching in tool planning methods were direct, significant and practical. A Tool Planning system would aid in checking the feasibility of production schedules, developing inventory control strategies and assist in tool budgeting. The following sections gives an overview of tool characteristics that can aid in establishing the fundamental principles on which a tool planning system could be built.

3.6 TOOL CLASSIFICATION

The ideal tool planning system would consider all types of tools. It is seen that different tools are used for different purposes and therefore have unique characteristics. Cutting tools are used to cut metals (eg HSS bars, inserts) and used on the machine until the end of the machining process, whereas non-cutting tools play different role. They could be used for setting up the process (set up tools) or holding the workpieces (eg jig & fixtures). Hence, it is essential to classify the tools on common attributes and study how such attributes would influence the Tool Planning methods.

The non-returnable tools (also known as 'disposables') have a very short life cycle in TM system. By definition, they are thrown away and not reused or returned to the stores. The management of such tools would be simpler than the returnables.

On the other hand, the returnable tools are retained by the tool stores upon satisfactory inspection of their condition. The returnables exist longer than disposables in any TM system and therefore, the planning of these tools becomes difficult. Very often, at any given time, the location and condition of returnables is not known. The transactions (issues/returns) at the tool stores take place more than once with returnables and the tool life normally decreases after each transaction. Therefore, better planning methods suitable in such dynamic environment are required for returnables.

The non-cutting tools can be either treated as returnables or can be permanently assigned to the workcentres. These would need different planning and allocation procedures.

Tools are normally grouped according to their functions for the purpose of selecting the required machining operations (drill for drilling, mill for milling), but for planning and control, they may need to be classified on different criteria. Many researchers classify tools to suit the requirements of their business needs [Galligan, Meister, Boyle, Kupferberg-1981, Melnyk-1988]. Some of the criterion that were used are listed below,

(1) Functional Classification : The tools are classified on the basis of the purpose for which they are built.

(a) Cutting tools

(a1) Returnables or reusables

(a2) Non-returnables or disposables

(b) Non-cutting tools, for e.g.

(b1) Set up tools

(b2) Jig & Fixtures

(2) "Cost/Volume Ratio : Some tools require tighter control than others. The "A-B-C" classification is a well known technique that help in controlling higher value tools tighter than lower value tools. The "A-B-C" classification based on the cost/volume ratio is explained in Table 3.1.

Table 3.1 : "A-B-C" Classification of Tools

CLASS	TOTAL COST	TOTAL VOLUME	EXAMPLE
'A'	80 %	5 %	Gear Cutters, Fixtures, Dies
'B'	15 %	15 %	Carbide inserts, standard mills
'C'	5 %	80 %	Standard Drills, Standard HSS bars

It can be seen that 'A' class tools carry higher value than 'B' class and therefore need tighter control. Savings on few 'A' class tools means significant savings in tool inventory costs. 'C' class tools have low values and therefore do not need greater attention. The limits for the 'A-B-C' classification are usually set on the basis of company's inventory policy.

(3) Flexibility of use : Some tools can be classed as either standard or special purpose, based on the range of their applications.

(a) Standard Tools :- Their design is standardised and they can be used for general purpose machining. Usually, they have greater flexibility in their use.

(b) Special Tools (Dedicated Tools) :- These tools are designed to suit specific machining requirements and can not be used for any other purpose. They are

also known as dedicated tools. They could be assigned either to the specific workcenters or products/ product groups.

(4) Procurement Lead Time : Tools can also be classified according to their procurement lead time. Although many tool manufacturers deliver orders within a few days, there are some tools with special requirements which can take longer than the standard delivery time. Those with higher lead time can become critical or limited resource and therefore would need advanced planning methods.

3.7 FUNDAMENTALS OF TOOL PLANNING

In repetitive manufacturing, the practice of using classical stock control techniques was common for all types of tools irrespective of their characteristics. Since different tools have different features, they require different planning approaches. There is no single rule that can be applied to plan and control all types of tools. This often led to excessive stock and obsolete tooling. Therefore, a hybrid approach of using a suitable technique for each type of tool is required.

The principal criterion of Material Requirement Planning (MRP) applicability is the existence of Master Production Schedule (MPS) to which raw material procurement, fabrication and subassembly activities are geared [Orlicky, 1975]. Similarly, a valid MPS is the prerequisite to execute any tool planning procedures. In the same way as

MRP is applicable to any discrete item purchased or manufactured that is subject to dependent demand, Tool Planning should be applicable to any tools purchased or fabricated in-house that is subject to dependent demand.

An approach similar to Production Planning can be considered for Tool Planning. The tool planning can be hierarchically structured similar to MRP system. Perera [1990] suggests such an approach applicable to FMS. This involves Aggregate Tool Requirements Plan (AGG-TRP) at the highest level indicating the effect of aggregate production plan on value and volume of tooling for long range planning. At the intermediate level, a rough cut capacity plan is suggested which considers only key tools to meet the requirements of provisional MPS. At the bottom level, Perera [1990] suggests a simulation based Tool Requirement Planning (TRP) running on MRP outputs. However, there is no relationship between AGG-TRP and rough cut tool capacity plan or between rough cut tool capacity plan and simulation based TRP. In other words, there is no vertical link between these tool planning activities.

This research proposes a hierarchical tool planning approach similar to the above but with only two planning levels, viz; AGG-TRP at the top level and TRP at the bottom level. The concept of rough cut tool capacity has been eliminated as the AGG-TRP can be used as a guideline for confirming the feasibility of provisional MPS. The proposed approach can be explained in fig 3.10. It can be summarised that a top-down production planning approach together with the bottom-up tool planning can be used to determine AGG-TRP as the final goal of the exercise. This approach has also a close horizontal interaction with the production planning hierarchy.

It is not feasible to make use of Perera's [1990] simulation based TRP for batch manufacturing. This is due to the lack of data needed for simulation, which is normally readily available in FMS executive computer. However, a TRP generated from the outputs of MRP has been used in this research. Further explanation on the tool planning methodology used can be made available from chapter 4 and 5.

The problem of tool planning can be partially resolved, if the total tooling requirements can be estimated on the basis of information available in MPS. The total tooling requirement can be defined as a set of three basic questions, viz; 'WHAT' tools, in 'WHAT QUANTITY' and 'WHEN'. Ideally, any Tool Planning system should be capable of answering these questions. Once the total tooling requirements are determined, then the tools can be planned in time buckets. This would give TM a better view of expected level of tooling activities over the respective planning horizon. This can also be termed as '*Off Line Tool Planning*'.

The information required from external sources to carry out tool planning procedures can be described with a diagram as shown in the fig 3.11 (off line tool planning) . The appropriate information can be obtained from Process Planning, Tool Engineering and Production Planning to provide answers to three major questions for tool planning.

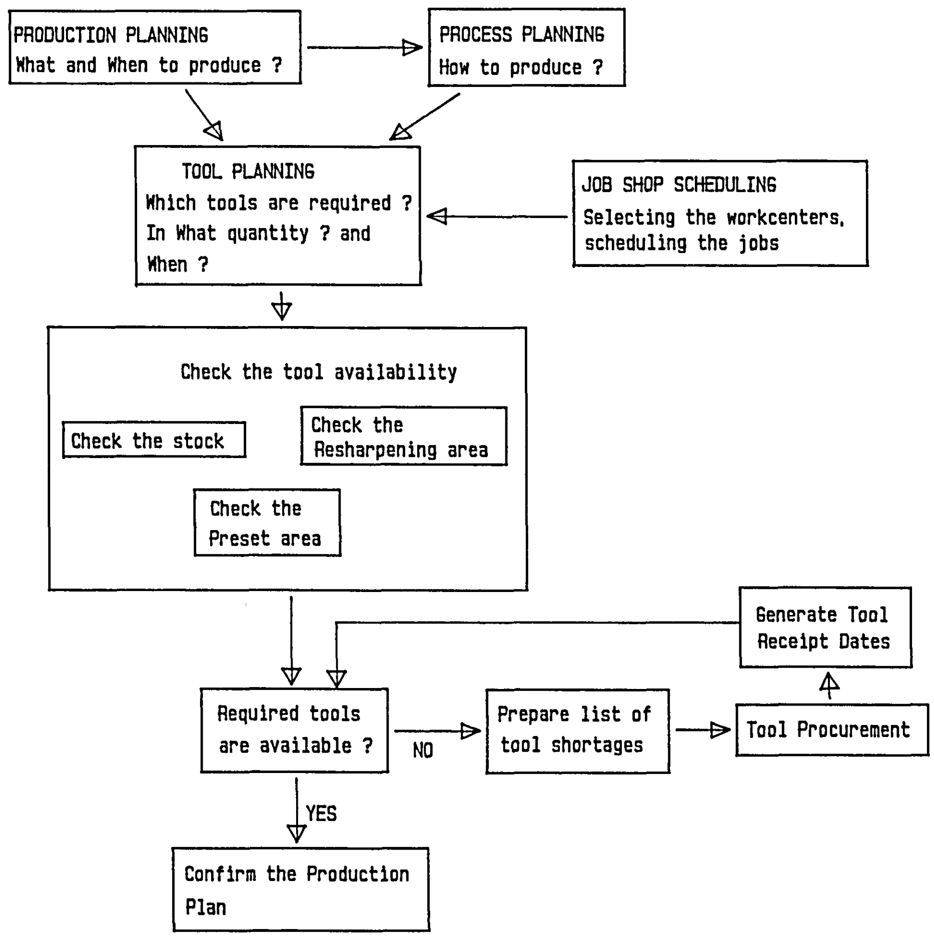
The process plans usually specify the sequence of operations and the respective tools required to perform them. Thus, exactly 'WHAT' tools are required can be found. The information such as tool life can be obtained from tool engineering (tool specification) details. This in conjunction with actual machining requirements can provide a rough estimate of tool life consumption. Thus, the tool quantity ('WHAT QUANTITY') can be known on the basis of the quantity of the components to be produced (which are at the lowest level of Bill of Material (BOM) structure).

The MPS gives a plan of end products to be produced in a time bucket. Once 'WHAT' and 'in WHAT QUANTITY' are answered, they can then be related to the end products from the BOM. Thus, the total requirements can be planned in the same sized time buckets as used in the MPS. This can also be termed as "*Aggregate Tool Requirement Plan*". The manufacturing lead time is taken into account to calculate the time 'WHEN' the tools are required. Once the tools are planned in time buckets, then they need to be allocated to either planned production orders or respective resources such as workcenters.

To achieve the objectives of tool planning i.e. determining tooling requirements and allocating tools, tremendous amounts of data need to be collected, processed and finally interpreted. The Data Flow Diagrams created earlier gives the data elements and data files to carry out the activities or functions. Such data files provide exactly what information is required to perform the tool planning procedures.

Fig 3.12 shows the changes in the status of tool as we come closer to the tool required date. This also describes the necessary tool management activities that are associated with the status of the tool. This research will be mainly focused on the first part, which is tool planning. However, the second part, i.e. tool allocation procedures (section 4.2.3) is not discussed in detail. The following chapter extends the discussion on tool planning and illustrates the proposed tool planning methodology.

FIGURE 3.1 COMPLEXITY OF TOOL PLANNING DECISIONS



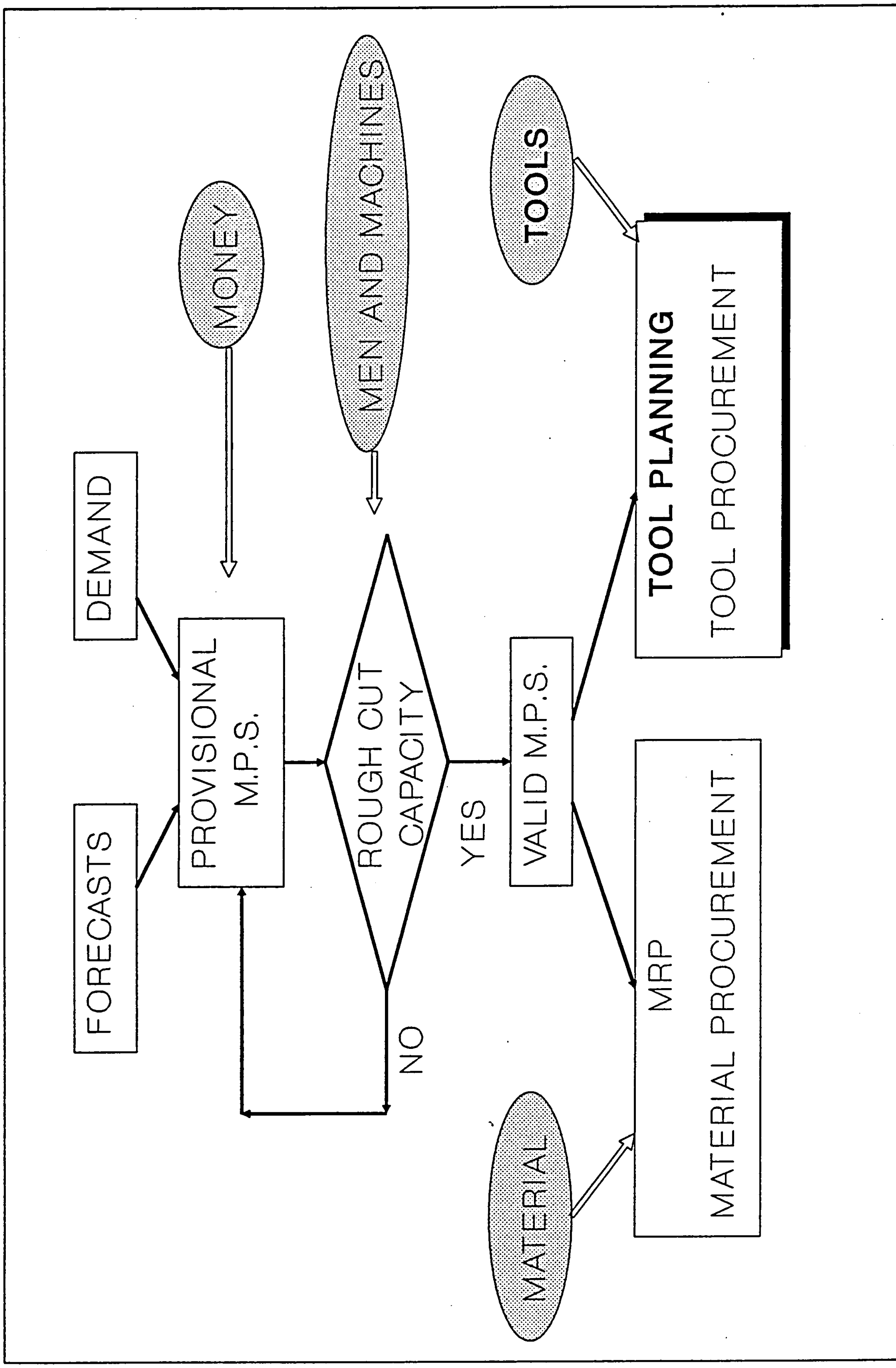


FIG 3.2 TOOL PLANNING - A COMPONENT OF MRP2

FIG 3.3 : DFD Diagram Conventions

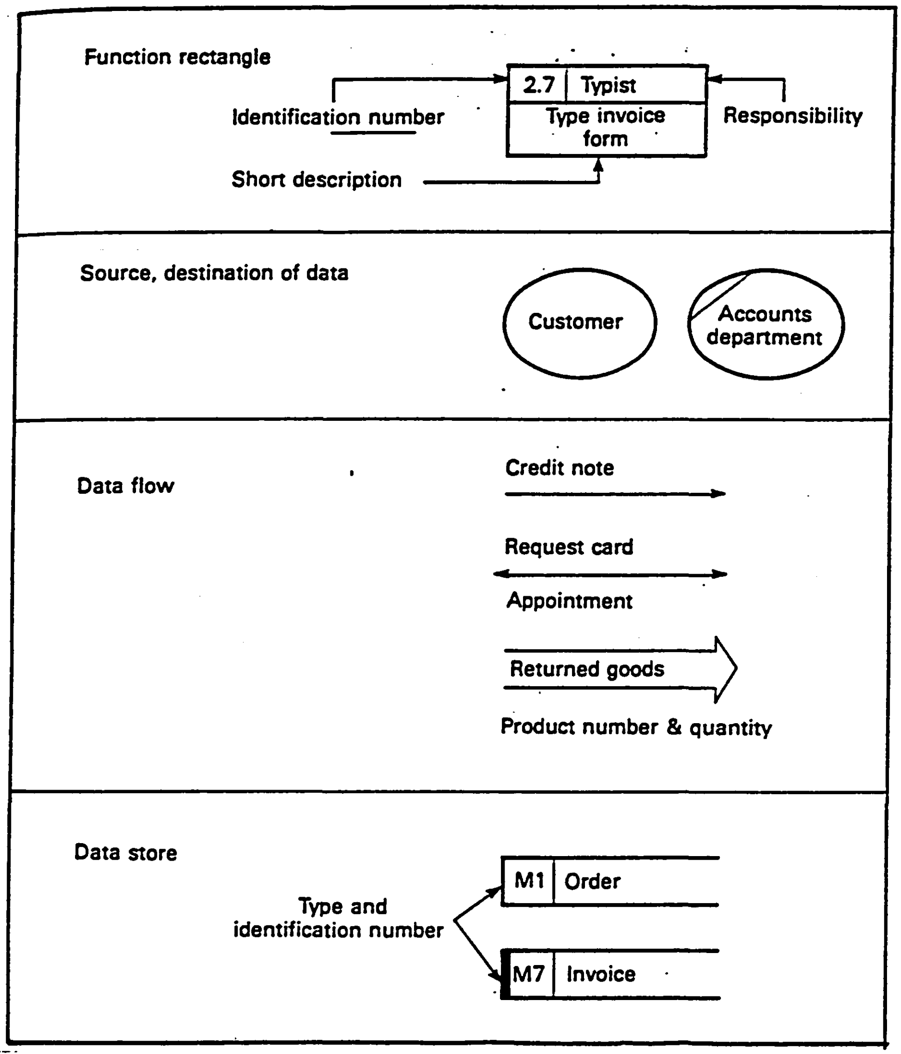


FIG 3.4 Functional Decomposition

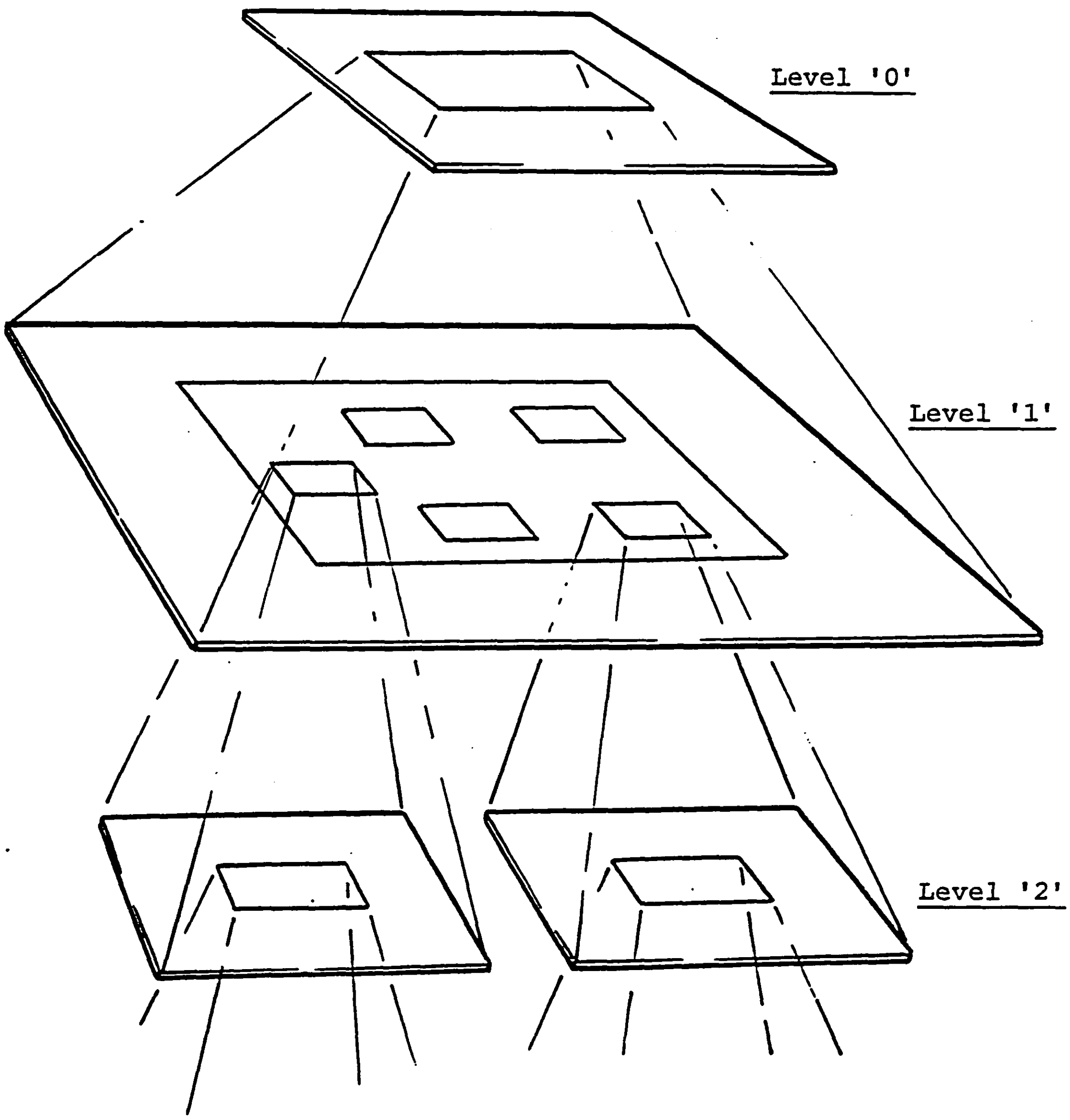


FIG 3.5 DOOR HANDLES

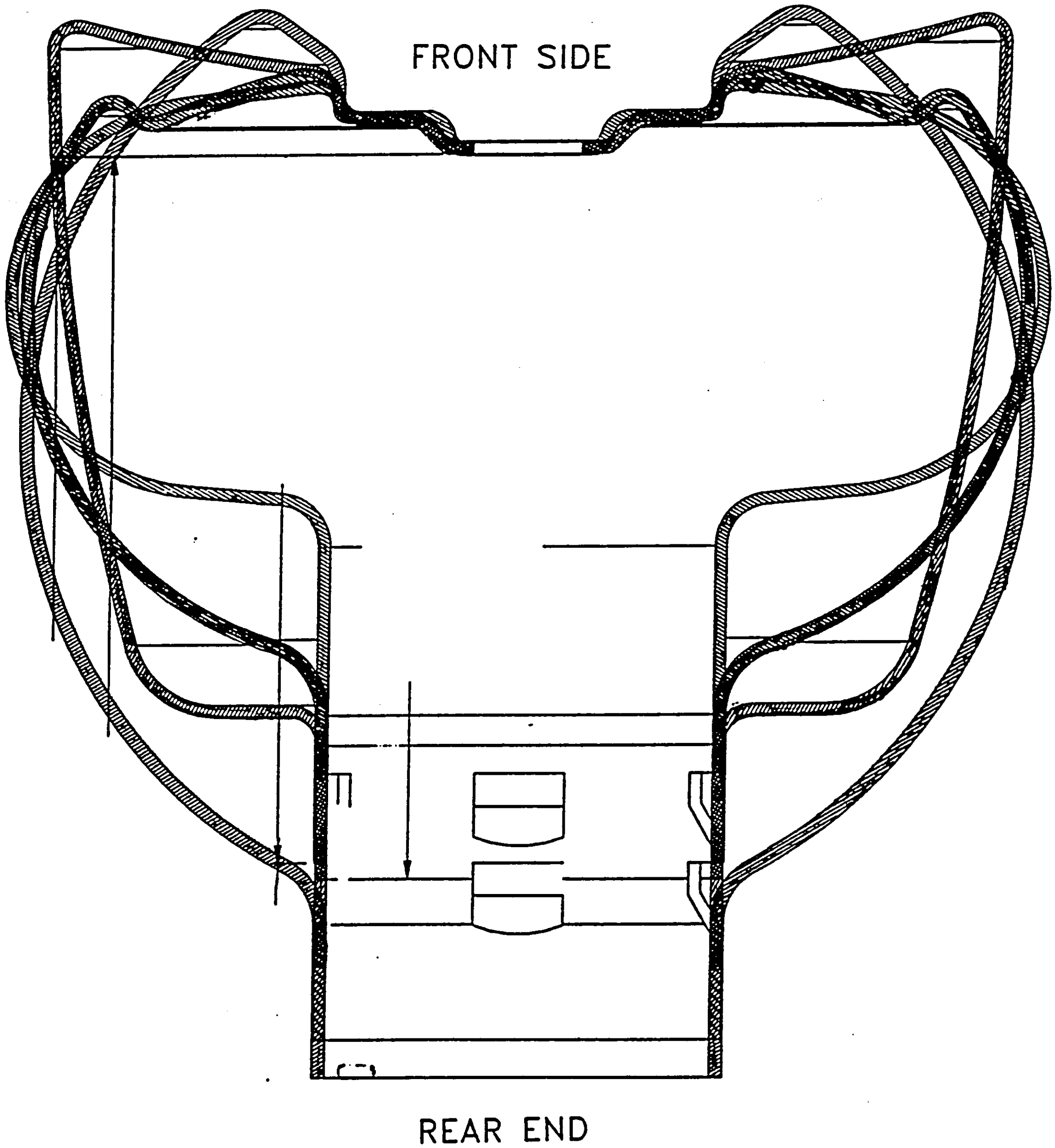


Fig 3.7. Level '1' DFD

Fig 3.8. Tool Stores Services. (Level 2)

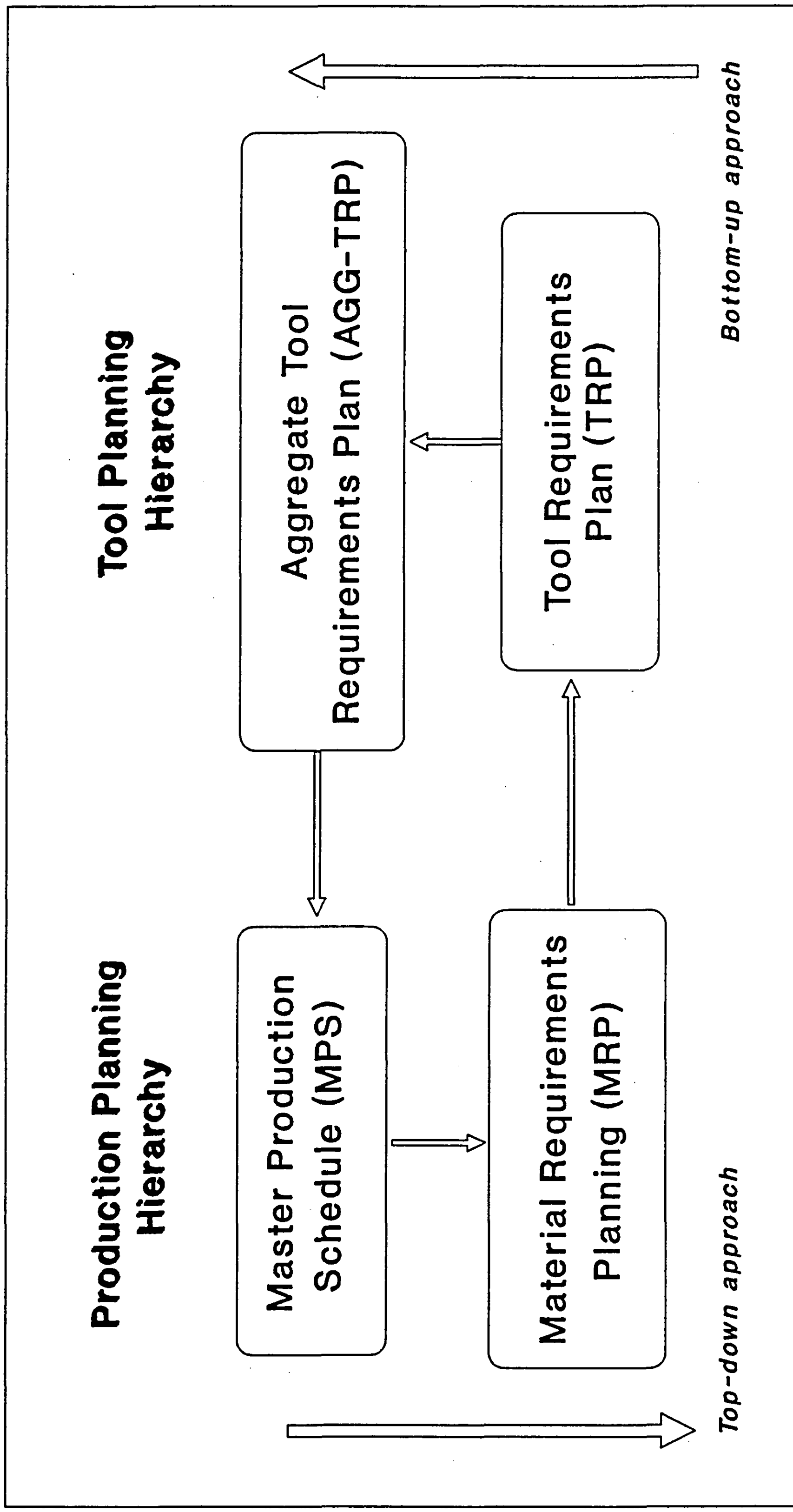


Fig 3.10 Hierarchical Planning Structure

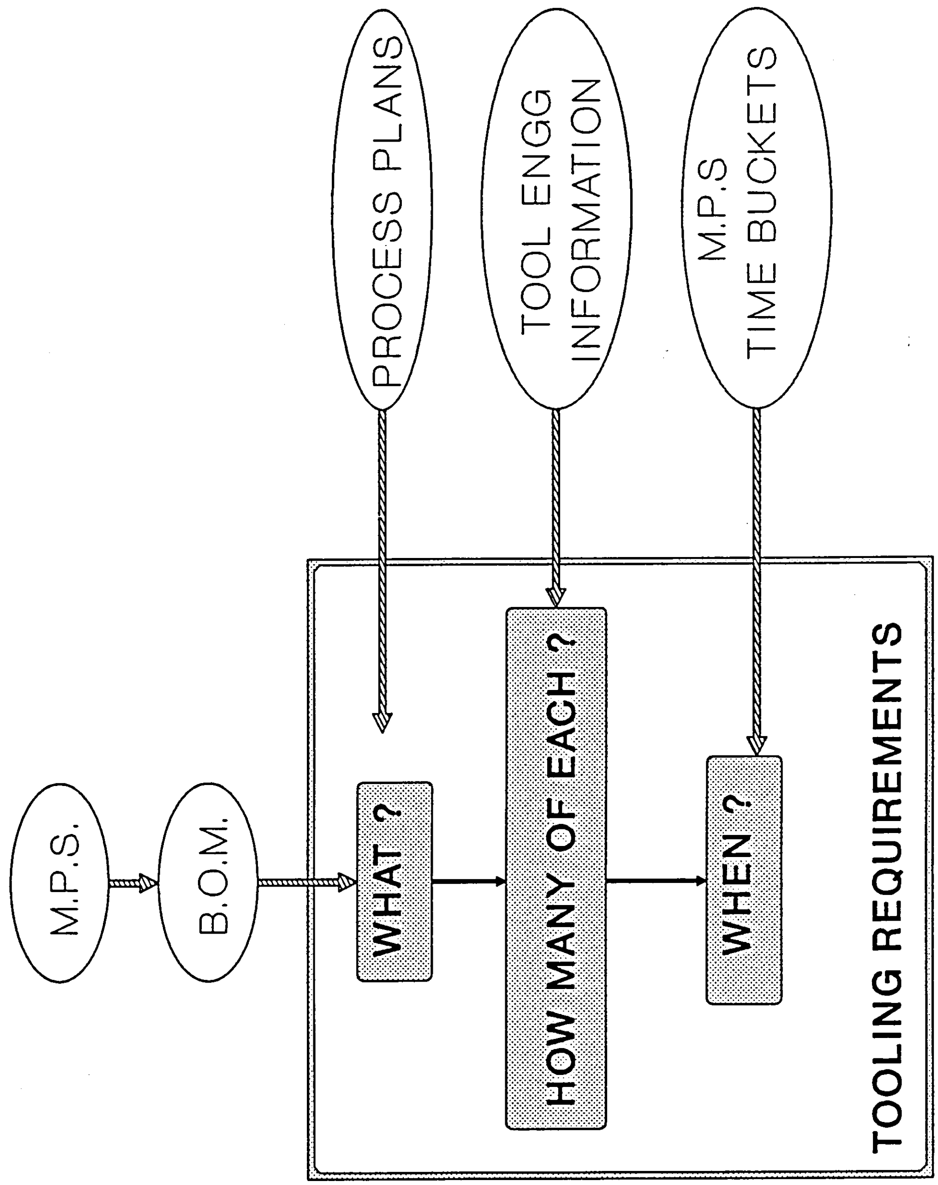
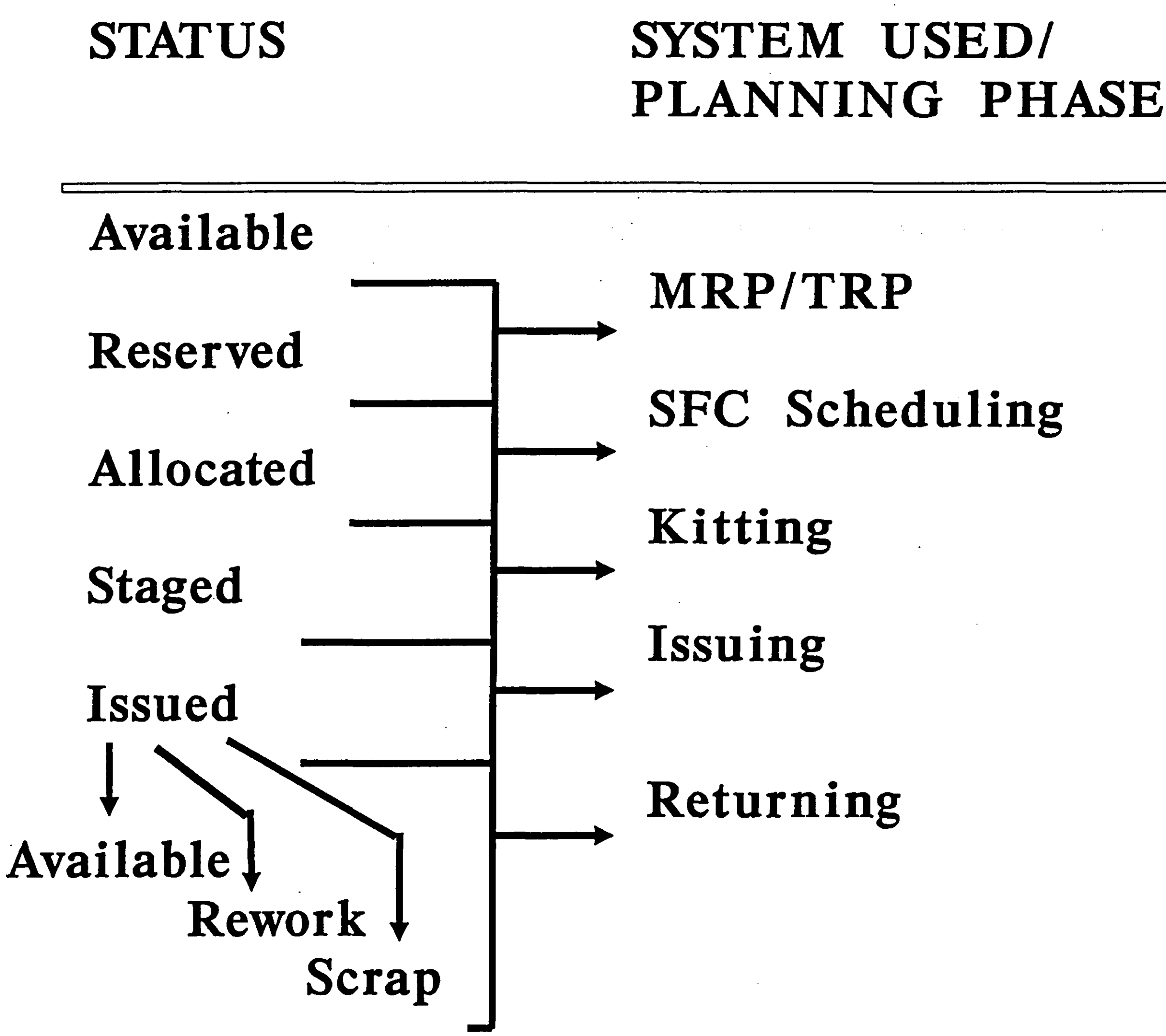


FIG 3.11 OFF LINE TOOL PLANNING

FIG 3.12 TOOL STATUS



4

TOOL PLANNING METHODOLOGY

4.1 INTRODUCTION

Having understood the fundamental principles of tool planning in the earlier chapter, it is important to see how a generic method for planning can be designed using these principles. The proposed Tool Planning Methodology based on the findings of the literature published by the various manufacturing professionals is laid out in detail in this chapter. The various stages of suggested planning activities are explained and the theory required for constructing a computerised tool planning system (TPS) is developed here. This chapter is concluded with the anticipated benefits and limitations of the proposed TPS.

4.2 TOOL PLANNING METHODOLOGY

The proposed tool planning methodology has been divided into three sequential steps.

Step 1 : Determine tooling requirements in time buckets based on information in MPS.

Step 2 : Generating purchase requisitions for tools (to be bought either from tool vendor or fabricated in-house) at different time phases and synchronize the tooling activities required for ensuring tool availability.

Step 3 : Allocate these tools to the respective production orders within the time buckets.

4.2.1 - Step 1 : Deriving the Tool Requirements in Time Buckets

The MPS stipulates the planned production (of end products) in either weekly or monthly time buckets. The MRP explodes the Bill of Material (BOM) to the lowest level of product structure. The Planned Order Release (POR) date of these individual components is set by back scheduling or by adding the respective manufacturing lead time. Thus, the MRP gives the POR dates and the net requirements of these components. Usually, certain types of components are bought out and others are either fabricated and/ or assembled in-house.

Each of these items have a unique process plan, as created by the process or production engineers. This gives the sequence of operations and the list of respective tooling required for those operations. The plan also gives the detailed information on the machining requirements and the type of workstation required. This answers 'WHAT' tools are required. The tools as identified from the process plan are linked to the tool information in 'Tool Master Database'. This database has a key field called 'tool code', which enable access to the various details about the tool, such as the 'A-B-C class', 'the procurement lead time', whether 'standard or special' tool and whether 'cutting or non-cutting' tool.

The next question 'HOW MANY OF EACH', would apply mainly to cutting tools. In case of non-cutting tools the problems are associated with availability and allocation more than knowing the exact quantity of each type of tool. The tool planning would attempt to answer only WHAT and WHEN for non-cutting tools. The synchronization of tooling activities with such tools are explained in steps 2 and 3.

Consider the cutting tools for calculating the tool quantity is required on a production batch size basis. In order to achieve this task, information such as estimated machining time of each operation and estimated tool life (in hours) of each required cutting tool is essential.

$$\textit{Tool quantity} = \frac{(\textit{Machining time}) * (\textit{Batch size})}{\textit{Tool life}} \qquad [5.1]$$

This gives a rough approximation of the number of tools required or the number of resharpening occasions of the tool. The issues related to the above equation are discussed in later sections in this chapter.

Aggregation of tooling quantity in time buckets :

Usually, the time phasing of MRP shows more than one component produced in a given time bucket. The tools that are common to these components need to be added in order to calculate the total tool quantity requirements. Thus, the entire range of exploded BOMs of different end products can be accounted and total tooling requirements can be accrued.

This can be explained by an example. Consider an end product 'A' which has the product structure as shown in the fig 4.1, 'B', 'C' and 'D' are the sub assemblies, whereas 'E' and 'F' are the components that are fabricated or manufactured in-house. The number with these alphabates (e.g. E-1) represent the quantity of each item required to make the assembly or sub assembly of items at higher level.

Let's assume that components 'E' and 'F' are required to be machined and cutting tools of consumable nature are involved in the process. The MRP explodes the BOM structure and determines the material requirements in time buckets (say weekly in this case). A typical plan is given in the Table 4.1. The components 'E and 'F' are produced from week #4 till week #8.

Table 4.1 : MRP in weekly time buckets.

	WEEK				
Component	**4**	**5**	**6**	**7**	**8**
A				1200	
B				400	
C			1560		
D		1530	1200		
E	1500	1200	2800	270	380
F	1490	4320	800	270	

As explained earlier, the Step 1 produces the tooling requirements of individual components. Once this is established, all the tooling requirements are scanned for common tooling. If more than one component with common tooling requirements are planned to be produced in the same week, then such tooling quantity is added. In this case, the quantity of common tooling for 'E' and 'F' is added for each of these weeks (i.e.; the week # 4,5,6 and 7) and the total requirements are calculated for product 'A'.

However, Whitney and Gaul [1985] have noted that the tooling constraint is not of the usual linear form. Tool requirements for the parts are not necessarily additive with the part types in batch. Different part types may use identical tools and can share those tools if the corresponding work is placed on the same machine, assuming those tools have some remaining useful cutting life [Amoako-Gyampah, 1992].

In this study, the above possibility is ignored for the purpose of simplification. Thus, all the end products like 'A' are taken into account one by one and aggregate requirements are determined on a weekly basis. Further explanation on data processing logic is available in chapter 6, where the prototype computer model is developed to demonstrate these planning concepts.

The aggregation of requirements gives a more accurate picture of tool consumption on a weekly basis derived from production plans. This could not only aid in procurement of tools but also give a better understanding of level of TM activities for the prescribed production plans. Such information can also be used for estimating the tooling budgets over that period.

Problems and Issues related to Step 1

(1) Problem of tool life estimation :-

It is not very difficult to extract information such as WHAT tooling will be required to machine the components. However, in order to calculate the quantity of tools (HOW MANY ?), there are some issues regarding the tool life estimates of certain types of tooling which has to be resolved.

According to Kuchinic and Seidmann [1988], the causes and behaviour of tool wear depends upon the cutting conditions and machining specifications of parts being machined. A tool is removed from service, once it starts producing the unsatisfactory parts or once it reaches its "economic tool life". An economic tool life applies to regrindable tools and disposable inserts.

Consider the two major categories of cutting tools as discussed earlier, viz; Returnables and Non-returnables (or disposables).

Returnables : The following factors makes it difficult to estimate the correct quantity of tools.

(a) It is difficult to estimate the remaining tool life on individual tools because the condition of the available tool at the time of planning would be different from the time at which it is planned for (due to its usage in the meantime).

(b) In case of multi tool set ups (such as turrets), individual tools have different tool life (this is the life available before the next resharpening event). This results into increase in tool change time (and hence increase in machine set up time). Thus, the methods of approximations used while calculating required tool quantity are very important.

Non-returnables : This problem could be simplified with consumable tools but only to some extent. According to its definition, the tool is disposed when the tool reaches its useful life. Many tool engineers have now realised the incomplete usage of such tools. In practice, consumable tools like disposable inserts are not returned to the tool stores for the tool life being not fully used. Considering these tool management practices, it is relatively simple to calculate the tool quantity of disposables on the production batch size basis.

(2) Other related issues in Step 1 :-

There are many other tools which can be ideally derived from the component process plans but need not require tighter control as their value (cost/volume ratio) is not significant. Although, some of these tools can be ideally defined as 'returnables', but they could be treated as consumables (eg HSS bars) for the planning purposes because they are classed as 'C' value tools. The major category of such tools are 'C' class items (eg HSS drill). The consumables or 'C' class tools that are used regularly can be planned using the following well known stock control techniques.

(i) Stock Replenishment

(ii) Reorder Point Techniques

(iii) Economic Order Quantity

(iv) Inventory Analysis and Categorisation by Function

(a) Cycle stock or Lot size inventory

(b) Reserve stock or Safety stock

(v) Aggregate Inventory Management [Orlicky]

The selection of appropriate technique depends on,

- The type of demand / rate of weekly consumption
- The procurement lead time
- Costs incurred due to stock-outs

The demand of each tool type varies according to its application. Therefore, it becomes necessary to adopt different stock control strategies for different tools. To gain better control over the tool inventory, it is essential to maintain a tool consumption database of such items. Additionally, the MPS may reflect some insight into the requirements of these tools. In repetitive batch manufacturing, a relationship between the tool consumption/usage and production volume can also aid in deducing the total tooling requirements of low value tools and consumables.

4.2.2 - Step 2 : Time Phasing of Tool Planning Activities

The objective of planning tools is to ensure their availability for the planned production. The MRP explodes the Bill of Materials (BOM) of the product structure level by level and determines the net requirements of individual components (or items) in the time buckets. The planned order release (POR) dates are set on the basis of the estimated manufacturing lead time of these individual components. Therefore, all the necessary tool management activities must be geared to the POR dates. Thus, the tool required date in effect, becomes the POR date of the component.

The required tool types as identified from the process plans can be grouped on the basis of tool classification explained earlier in Chapter 4 (section 4.2). The special tools (or 'A' class and or critical tools) with high procurement lead time can be segregated from 'B' and 'C' class standard tooling.

The aim is to plan and order tools at two different time phases. Fig 4.2 shows the time phasing of various tool planning and related activities that are geared to POR dates. The first phase would consider special 'A' class tools from the first category at least two to three weeks before the POR date (depending upon their procurement lead time). The second phase would plan the activities for second category of tools approximately one to two weeks before the POR date.

The tools such as jigs and fixtures can be associated either to a product type (or product family) or to a workcentre. Such tools could be allocated to the respective resources and scheduled for the required time.

Tool Planning Phase I :-

The first category of tools which are either 'A' class tools, or special tools with high procurement lead time or the ones which are not frequently used are considered to be the most important tools. Different tools have different procurement lead time. Say for example, they are classified into three types, Procurement Lead Time =

(a) $>$ 14 days

(b) $>$ 2 days but $<$ 14 days

(c) $<$ 2 days

(The above method of classification would depend upon the individual requirements of the business. The above figures are chosen to illustrate the concept only and would depend upon individual company's planning strategies.)

Type (a) and (b) tools are either

- Stocked before hand- if the component is produced regularly and usually on 'make-to-stock' basis. OR

- Purchased after receiving the production order- if the component is produced rarely and on 'make-to-order' basis.

Traditionally, the factors that influence the tool stocking strategy are shown in Fig 4.3. Table 4.2 shows how the various factors (such as class of tools, procurement lead time, and whether standard or special purpose tool) would influence the stocking or planning strategies. The choice of the appropriate strategy is left to the planner, which would vary from one operating environment to another. Table 4.2 provides the foundation on which the planners can make the decisions regarding tool stocking or ordering policies.

Table 4.2 Tool Planning Strategy Selection

CLASS	PROCUREMENT LEAD TIME-DAYS	STANDARD OR SPECIAL	PLANNING STRATEGY
A	>10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	>2 AND <10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	<2	Standard	Stocking/Ordering
		Special	Stocking/Ordering
B	>10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	>2 AND <10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	<2	Standard	Stocking/Ordering
		Special	Stocking/Ordering
C	>10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	>2 AND <10	Standard	Stocking/Ordering
		Special	Stocking/Ordering
	<2	Standard	Stocking/Ordering
		Special	Stocking/Ordering

If they are stocked, then a greater degree of sophistication is required in forecasting the demand. A small variation in estimating the requirements would increase the tooling costs significantly. The problem of forecasting the demand becomes more complex when such tools are shared by two or more workcentres.

If the second policy of purchasing these tools after the receipt of orders is adopted, then the purchasing activities must begin at the appropriate time (i.e. two to three weeks before planned order release {POR} date). The necessary prerequisites to use this approach are,

- Less volatile MPS
- The exploded MRP with wider planning horizon.

Tool Planning Phase II :-

As we move closer to the POR date, a second tool planning review could be held. This would take place approximately one to two weeks before the POR date. The following tools would be taken into account,

- all 'A' and 'B' class tools with lead time less than 2 days and
- all those purchased in Phase I

The commonly used 'A' and 'B' class would be stocked and the net requirements could be calculated.

$$\textit{Net requirements} = \textit{Total required - Total available}$$

The purchase orders could be released for the net requirements and the available tools to be allocated and scheduled for that component in that time bucket. Other activities such as tool preparation and tool issue could also be synchronised according to their sequence and the activity time.

4.2.3 - Step 3 : Allocation of tools

This step in planning is carried out once the tools are procured. The earlier stages accomplish the tasks of determining the tooling requirements and organising the related tooling activities geared to the tool required date (or POR date of the component). Tool allocation is considered to be an inherent and critical element of the dynamic production planning problem and has a significant impact on the performance of the manufacturing system [Veeramani]. Therefore, it is important to have effective methods of allocation. This topic is outside the boundary of this research and hence it is not discussed in detail.

4.3 REQUIREMENTS FOR RUNNING A TOOL PLANNING SYSTEM (TPS)

In order to run the proposed tool planning system, tremendous data is required in appropriate format to carry out the necessary data processing. As explained earlier, tool planning acts as a bridge between production planning and TM system. Therefore, the existence of a formal MRP system becomes essential to run the proposed TPS. TPS demands the data from the MPS to derive the requirements. The entire data processing depends upon the information contained in the MPS.

The process plan database should specify the tool required for operations. Similarly, the tool engineering database should be able to keep records of tool life for cutting tools under normal machining conditions. The cutting conditions such as the cutting speed, the material of the component being machined, the material of the cutting tool vary from one operation to another. Therefore, it is difficult to estimate the tool life. An attempt has been made to estimate the tooling quantity and the aggregate requirements based on the simplified tool life estimation, which takes no account of different materials and cutting conditions.

In order to calculate the net requirements, the information on tool quantity in stock should be known. Which means that a well maintained database of tool transactions is also required from tool stores. This should also provide information on the location of the tools ('WHERE' the tools are). The literature on TM indicates this problem as serious and is found very commonly with most manufacturing industries [Mason, 1988].

4.4 BENEFITS AND LIMITATIONS OF TOOL PLANNING SYSTEM (TPS)

The benefits of the proposed Tool Planning System are,

(1) tool requirements are based on the batch sizes of components and not the tool forecasts. Therefore, fluctuations in volume of production does not affect the tool availability.

(2) reduction in tool inventory by minimising hot purchases and utilising tools effectively by allocating them to appropriate resources.

(3) minimising production stoppages by synchronising TM activities like kitting, preparation and issuing.

(4) better understanding of tool consumption on the basis of their application and value. This data can also be used for preparing tooling budgets periodically.

However, the TPS impose some limitations which are given below.

(1) It can not be used if the MPS is very volatile.

(2) It is not suitable for heavy fabrication type of environment. If the part type variety is high, then it is difficult to determine requirements at aggregate level.

(3) It gives only a rough estimate of tooling requirements.

(4) Wider the planning horizon better the plans.

The following chapter explains how the suggested TPS is built using a database management system.

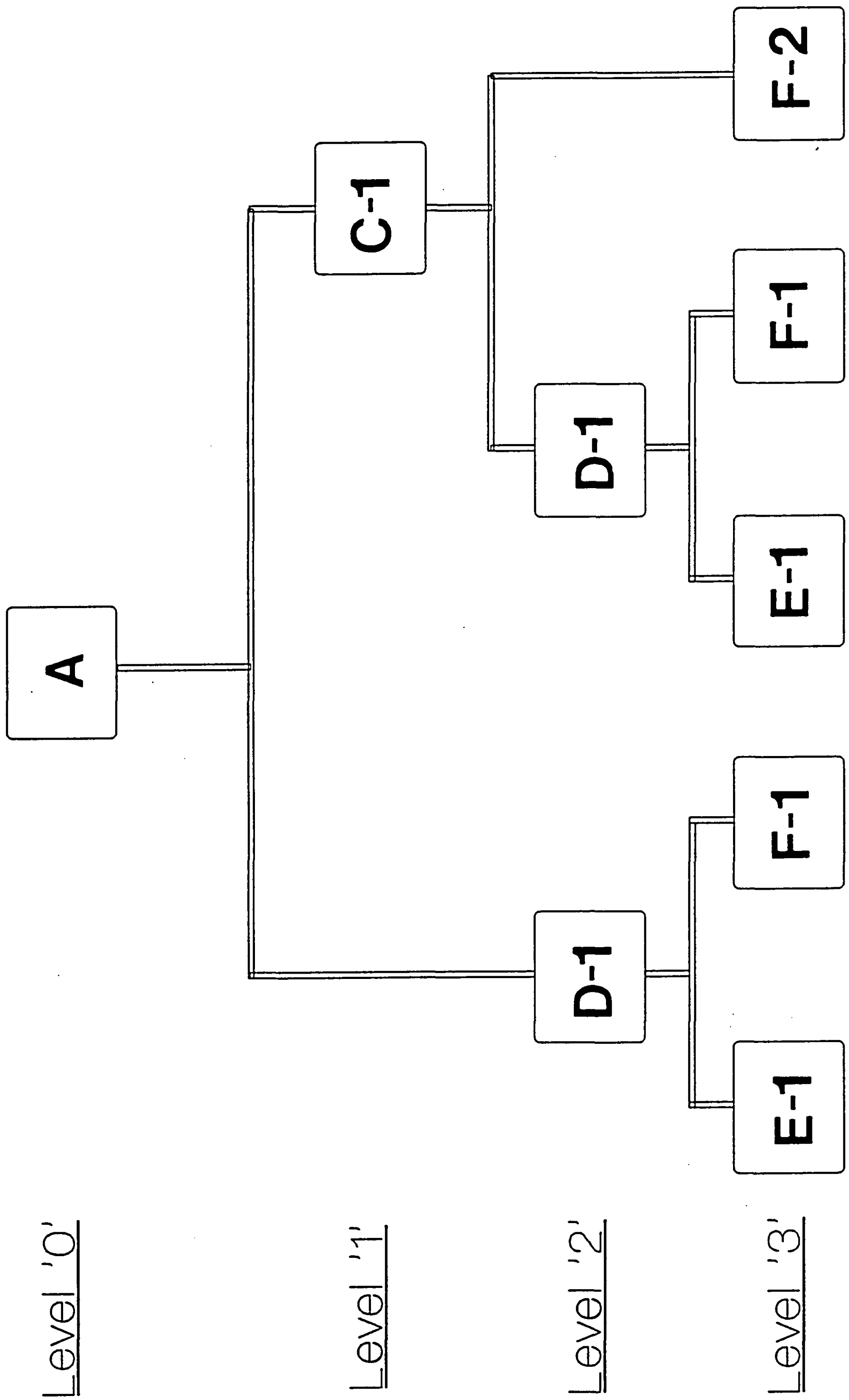
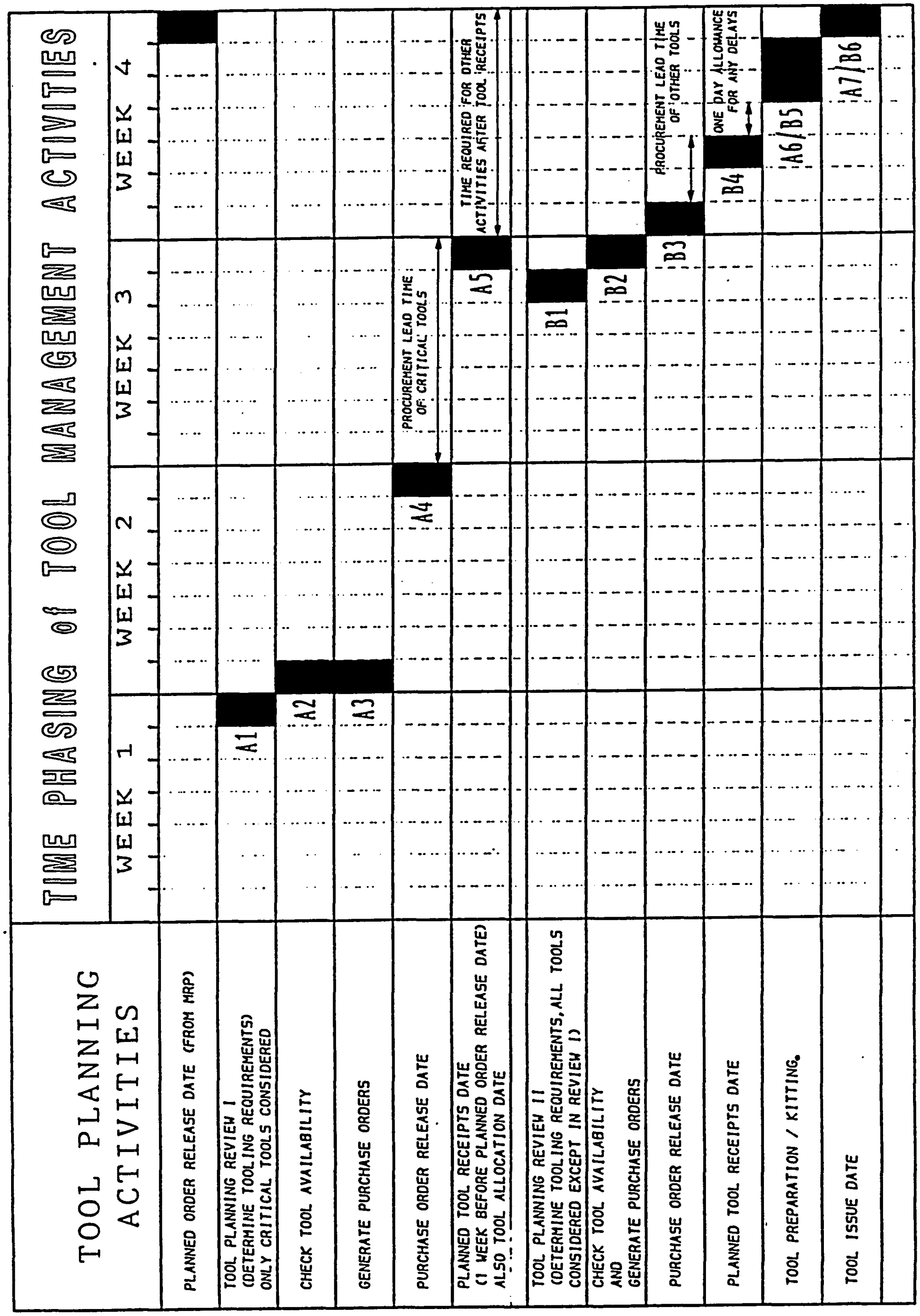


FIG 4.1 - PRODUCT STRUCTURE OF END PRODUCT 'A'

Fig 4.2 Time Phasing of TM Activities



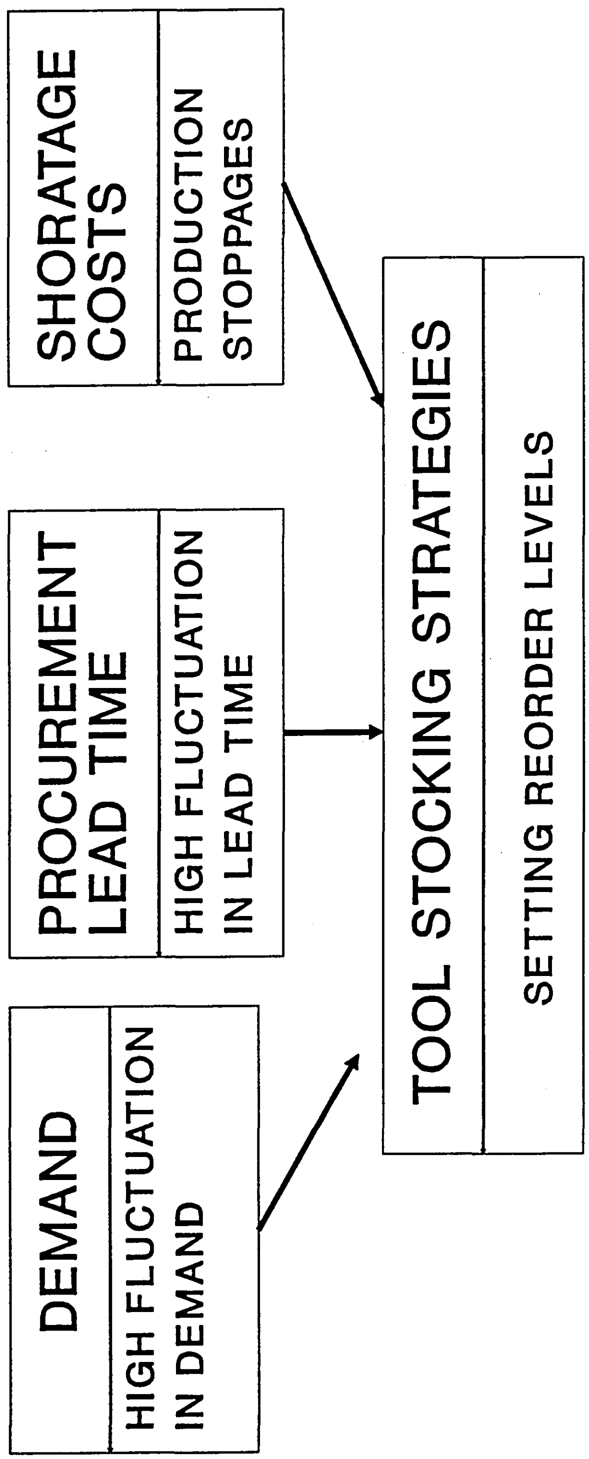


FIG 4.3 FACTORS INFLUENCING TOOL STOCKING STRATEGY SELECTION

5

DEVELOPMENT OF TOOL PLANNING SYSTEM

5.1 INTRODUCTION

The earlier chapter illustrated the theoretical aspects of the suggested tool planning methodology. This chapter describes how such a system is developed using the computer. The proposed Tool Planning System (TPS) has a very complex data processing logic. The TPS requires very high volume of data to be handled. Therefore, the usage of a computer becomes necessary for this work.

The entire work has been divided into two modules. The first module, which is the TPS, determines tool requirements on the basis of the MRP outputs and stores them in weekly time buckets. The second module consists of a simulation model that reads data from the TPS output, simulates the production operations and generates appropriate reports. Both the modules are coded and merged using the database package called Foxpro (Version-2). This chapter describes how the computerised TPS was developed, whereas, the simulation model is explained in Chapter 6.

The reason for developing the simulation model was to test the effectiveness of the suggested TPS. In order to maintain the integrity of simulation modelling and the TPS, it was decided to develop both the modules on a single DBMS. This avoided the complexity of interfacing two systems built on different platforms.

The overview of various database management systems (DBMS) is given here, and the reasons for selecting Foxpro-2 is also justified. This chapter extends the discussion to the detailed analysis of data, database specification and the programming logic built in the TPS.

5.2 SELECTION OF A SUITABLE DATABASE PACKAGE

Usually, every Tool Store stocks a large number of tools. Therefore, a good TM system would require a large size database with a number of tool records for storing the individual tool information. Hence, it is important that a TM system is developed using a database management system (DBMS), which could generate the reports for TM decision making processes.

Similarly, the tool planning function requires various databases to be accessed to obtain appropriate information for data processing. A good DBMS is a prerequisite for the success of any tool planning system.

There are two types of PC database, flatfile and relational. The flatfile type is designed for a single user wanting to record and retrieve one type of data. Relational databases are a hierarchy of flatfile tables and therefore, are suited to interdepartmental or interuser needs. So far, the PC database market was dominated by Ashton Tate's dBase products, such as dBase-III+ and dBase-IV. Although sophisticated enough to manipulate server data, is by no means the best choice to build the applications such as Tool Management. Today's PC market offer equally sophisticated products like Dataease, Paradox and Foxpro. All these DBMSs offer the higher database standard, which is the Structured Query Language (SQL - was originally a standard for mini computers and mainframes).

It was decided to choose dBase IV due to its distinct advantages such as the SQL server, the new control structure (commands like 'SCAN...ENDSCAN' which establishes a loop to find and process records which meet a specific condition) and elegant features like user-definable menus and window control. Upto twenty windows can be opened on the screen at any one time.

During the initial stages of development of the proposed TPS, dBase-IV was used. The system was running successfully initially, but with the addition of new database files during the process of improvisation, dBase-IV could not handle the volume of data that had to be processed (where the system is considered to be only prototype, imagine the volume of data for a fully fledged system). One of the reasons being that dBase-IV could not open more than ten database files at one time. As a result the program could not be run successfully and hence the need for a better PC based DBMS system was clear.

Fox Software has been producing the Fox range of improved dBase type software since 1984. Foxpro (Version 2) is a DOS based DBMS and has a mouse driven pop down interface. Just like dBase, it can be controlled from the menu system, from a command line or from a program file. The interface is much more attractive and intuitive than dBase's. One of the computer surveys in the U.K. show that Foxpro is at least three times faster (this includes mathematical and searching speed) than dBase IV and more importantly could open more than ten database files at any one time of data processing [Liardet and Whitehorn, 1991]. These advantages of Foxpro over dBase-IV made Foxpro an obvious choice for further development work.

5.3 DATABASE SPECIFICATION FOR TOOL PLANNING SYSTEM

The functions of tool planning system (TPS) were established using Data Flow Diagrams (DFDs). The analysis of the generic structure of entire Tool Management is explained in Chapter 3. The analysis stage also defines the sub-functions such as, Tool Stores Services, Tool Engineering Control etc. of which Tool Planning was chosen for further understanding and the development of techniques.

The operating logic of the proposed TPS was defined only to a limited extent at the analysis stage in the sense that only 'WHAT' a TPS should achieve rather than 'HOW' it could be done. The later sections in this chapter explain 'HOW' the operating logic is developed based on the principles of tool planning as described in Chapter 4. This logic shows exactly how the data is processed and it also lists the input and appropriate output.

The development of TPS is limited to such an extent that only a prototype system is aimed to be produced. It was recognized that a prototype system would be sufficient to demonstrate the concept of a tool planning mechanism. In practice, a full fledged system may be required and can be built based on the principles which are laid out in this chapter.

The various data attributes required to carry out the essential functions of TPS were laid out in the DFDs in Chapter 3. It also gives the appropriate data stores created for the TPS. This information from the DFDs is useful in the next stage of the analysis, called as 'Data Analysis' in Software Engineering terms. The various data attributes were listed and linked to the appropriate data stores. The data store only represent a set of data attributes that a database file would contain. However, the DFD does not give any idea about the relationships between the various data attributes. This study is carried out here.

The task of Data Analysis (Normalisation) and the design of the programming logic is carried out simultaneously. The assumptions and the limitations of all the two modules (the TPS and simulation system) are given in chapter 6.

Normalisation (Data Analysis) :

A database is a file of structured data stored in a computer but arranged so that they can be accessed in many different ways for use in different applications. The idea is that the same data is stored only once but can be manipulated by the database management system (DBMS), so that data files can be shared by different pieces of software [Samways, 1989]. In other words, it is "a collection of non-redundant data shareable between various application systems."

In order to design a correct, consistent and stable database, a technique called "Normalisation" is used. Normalisation is a method used for transforming complex data structures into simple tables which are in their third Normal Form (3NF). The third normal form is defined as "the process of eliminating functional dependency between non-key attributes of the data structure" [Howe, 1983].

Normalisation of data structures is necessary to ensure that they are represented in their simplest form and also remove the possibility of loss of data integrity. The simplest way to reduce the incidence of inconsistent data is to eliminate unnecessary duplication of data [Howe, 1983].

The DFDs created in chapter 3 give a list of data store interacting with the TPS functions. The data attributes from the DFDs were then identified and only those data attributes which are required for the TPS were considered for data analysis. The relationships between these data attributes formed a guideline for designing the number of fields required in a database file. The ones closely related to each other and regarded as the primary and composite keys became a part of one database file.

The process of Normalisation is explained with an example in the following sections. For e.g. tool is considered as an entity with several data attributes as shown in the following table.

TOOL
Tool Code
Tool Name
Tool Identification Code
Estimated Total Tool Life
Unit Tool Life
Available Tool Life
Returnable or Non-returnable
Procurement Lead Time
Standard or Special
Class (A/B/C)
Price (£££)
Tool Size
Tool Material
Current Stock
Minimum Stock
Tool Supplier

In the process of first normal form , the repeating data attributes are grouped. Each of this group is given a name (which eventually becomes the name of the database file). The grouped attributes and their names are given in the following table.

TOOL	REPEATING GROUPS OF DATA ATTRIBUTES
Tool Code	
Tool Name	
Tool Identification Code	Tool Life Attributes
Estimated Total Tool Life	
Unit Tool Life	
Available Tool Life	
Returnable or Non-returnable	Tool Characteristics
Procurement Lead Time	
Standard or Special	
Class (A/B/C)	
Price (£££)	
Tool Size	Tool Engineering Specifications
Tool Material	
Current Stock	Tool Stock Details
Minimum Stock	
Tool Supplier	Tool Supplier Details

The key attributes such as tool code and tool identification are identified, based on which other dependent attributes can be identified. For e.g. attributes such as Current Stock and Minimum Stock are dependent on the tool code. Therefore, the stock attributes are said to be functionally dependent on the key attribute 'tool code'. In this manner, the key attributes and the dependent attributes are put together in one table. The resultant grouped data attributes can be represented in the following tables.

Tool Characteristics
Tool Code
Tool Name
Returnable or Non-returnable
Procurement Lead Time
Standard or Special type
Class (A/B/C/)
Price

Similarly, the other groups such as Tool Life, Tool Engineering Specifications, Tool Stock and Tool Supplier Details are tabulated. They can be represented as,

Tool Life Attributes
Tool Code
Tool Identification No.
Estimated Total Tool Life
Unit Tool Life
Available Tool Life

Tool Engineering Specifications
Tool Code
Tool Size
Tool Material

Tool Stock Details
Tool Code
Current Stock
Minimum Stock

In the second normal form, the task is to ensure that the there is no functional dependencies among the non-key attributes. For e.g. In the above table of Tool Stock Details, the key attribute is the Tool Code whereas, the Current Stock and the Minimum Stock are the non key attributes. Therefore, both the non key attributes should be independent of each other.

In the final task, the composite keys are identified. For e.g. in the Tool Life table, there are two keys required in order to access the information on tool life. Tool Code and Tool Identification Number are the attributes that are required to be known before any further information on that particular tool can be obtained. Such keys are known as Composite Keys. In this manner, all the grouped data attributes are represented in their 3rd Normal Form (3NF).

The next task was to analyze these data groups and study the interrelationship between them. This was established by defining exactly how the data would be accessed for processing (for either viewing or modification). This also enabled the specification of the format of the input and the output. All the above issues assisted in defining the structure of database files.

The configuration of all tool planning database files and all the data attributes in their third normal form are explained below. The actual codes for the databases file names and the data attributes are given in brackets. The file extension '.dbf' stands for 'database file'.

Production Plan Database (PART_ORD.DBF) : This is the production plan of individual components after the explosion of the product structure to the lowest level by the MRP system. It is assumed that this information will be generated and furnished by the MRP system to the TPS. The structure of the database file is shown below,

Production Plan Database (PART_ORD.DBF)

Primary Key	*Data Attributes*
Order Number (order_no)	Part Number (part_no)
	Batch Size (batch_size)
	Planned Order Date (plord_date)

Process Plan of a component 'Part number-1000' (P1000.DBF) : It is assumed that the process planning would provide the information in the appropriate form as desired by the TPS. This is the key area where the required tooling is identified on the basis of machining requirements. Information on manufacturing lead time is also accounted here, so that the machining content could be estimated and the tool life consumption can then be studied.

Each component in the system has a separate database file representing a process plan.

For the purpose of convenience, the parts are designated as 'P1000' for part number 1000, 'P1001' for part number 1002 ..and so on for the subsequent parts in the system. The various data attributes contained in such a process plan can be stated in their 3NF in the following manner.

Part Number Database

Primary Key	*Data Attributes*
Operation number (opn_no)	Operation name (opn_name)
	Workcenter number (wcent_no)
	Tool code (tcode)
	Set-up time (set_time)
	Machining time (run_time)

Tool Characteristics Database : This database stores all the information on tool characteristics, such as, whether it is returnable or consumable, standard or special, the type of class (A-B-C) and the procurement lead time. This database greatly helps in classifying tools on different grounds and treat them differently for planning purposes.

Tool Characteristics Database (TCODE_NA.DBF)

Primary Key	*Data Attributes*
Tool Code (tcode)	Tool Name (tool_name)
	Returnable ? (returnable)
	Procurement Lead Time (proc_lt)
	Standard ? (standard)
	Class (class)
	Price (price)

Tool Engineering Database : This database stores the engineering details of tools. In practice, there is much more on tool engineering specification, but the proposed TPS considers only the limited information that is required for processing. The tool code is the primary key for the access. The structure of the database is given below.

Tool Engineering Database (TOOL_ENG.DBF)

Primary Key	*Data Attributes*
Tool Code (tcode)	Tool Size (tool_size)
	Tool Material (tool_matl)

Tool Life Database : This database was created to store all the information related to tool life. Each tool type has a specific estimated life as given by either tool engineering control or by the tool manufacturer (the estimated life is under normal cutting conditions). A unit tool life can be defined as the life consumed just before it is withdrawn from its normal use (either for reconditioning or for disposal). In the TPS, each tool is considered to have certain tool life. In practice, to what extent it is feasible to obtain this data is still a debatable issue. However, it was assumed here that such information would be available with the advent of sophisticated tool engineering technology.

The life available on each tool is represented by 'ava_life'. The 'no_regrind' is used for keeping the records of number of times each tool was withdrawn for either regrinding or reconditioning. The TPS assumes that each tool is limited to certain number of regrinding events, beyond which the tool needs to be disposed. The disposables (non-returnables) do not have these attributes. The Tool Code and Tool Identification Code form the composite key for access. The structure is represented in the following manner.

Tool Life Database (TOOL_LIF.DBF)

Composite Key	*Data Attributes*
Tool Identification Code (tool_id)	Estimated Life (est_life)
Tool Code (tcode)	Unit Tool Life (unit_life)
	Available Life (ava_life)
	Number of Regrinding Events (no_regrind)

Tool Inventory Database : The tool store normally keeps the records of tool transactions and this is performed using this database file. The information on current stock level (in_stock), the minimum required stock (min_stock) and the order size (order_size) are stored here. These stock details can be obtained using tool code (tcode) as a primary key. In chapter 8, where the simulation study is carried out, it is discussed how these stock values would affect the production and tool management activities. The structure of this database is given below, *Tool Inventory Database (TOOL_INV.DBF)*

Primary Key	*Data Attributes*
Tool Code (tcode)	Current Stock Level (in_stock)
	Minimum Stock (min_stock)
	Purchase order size (order_size)

Weekly Requirements Database : This is the one of the output files of the TPS. The required tool types (tcode) are put in the appropriate weekly time bucket which are represented by the week number (e.g. week1, week2 .. and so on). The TPS accounts for ten weeks of which only four weeks are used for output analysis (four weeks plan is a typical example in most industries). However, a full fledged TPS can have as many weeks as desired. The required tool quantity can be accessed by the tool code, which is a primary key in this case. The structure is simple and can be shown in the following manner,

Weekly Requirements Database (WEEK.DBF)

Primary Key	*Data Attributes*
Tool Code (tcode)	Week1
	Week2
	Week3
	Week4

Tool Aggregate Database : The process of aggregation involves adding the tool quantities that are common between the components which are released in the same time buckets. The detailed explanation on tooling aggregation can be made available in Chapter 4.

This database purely acts as a temporary record file holding the tool code and the required quantity against it. The records are verified for commonality and reqd_qty is added if it exists. All the records are deleted after each run of TPS. The tcode acts as the primary key and the structure is shown below.

Tool Aggregate Database (TOOL_AGG.DBF)

Primary Key	*Data Attributes*
Tool Code (tcode)	Required Quantity (reqd_qty)

Tool Requirements Database : The details on tooling requirements based on individual machining operations are temporarily held in this database. The data form this database is processed partially and then it is other part is transferred on to the other database files such as ALL_REQ.DBF for further processing. Operation number forms a primary key for access. The structure is given below.

Tool Requirements Database (TOOL_REQ.DBF)

Primary Key	*Data Attributes*
Operation Number (opn_no)	Tool Code (tcode)
	Tool Required Date (reqd_date)
	Tool Required Quantity (reqd_qty)
	Estimated Tool Life (est_life)
	Operation Time (run_time)

All Requirements Database : This database acts as a temporary file for data processing. The data retrieved from the various files is processed and stored in a required format for further processing. The main inputs to this file being the data from 'PART_ORD.DBF', 'P1000.DBF' and 'TOOL_REQ.DBF'. Before commencing every run in the TPS, all the old database records are deleted for the new ones to be stored. The requirements are determined on the basis of Order Number. Order Number together with Operation Number form a composite key. It is seen that this database contains the same fields as some other databases such as 'wcent_no', run_time (both found in process planning like 'P1000.DBF'). The structure can be represented in the following manner.

All Requirements Database (ALL_REQ.DBF)

Composite Key	*Data Attributes*
Order No. (order_no)	Workcenter No.(wcent_no)
Operation No. (opn_no)	Tool Code (tcode)
	Required Quantity (reqd_qty)
	Operation Time (run_time)

The relationship between the various databases created in the above manner can be diagrammatically represented in Fig 5.1. The arrows indicate how data attributes flow from one table to another. There are several other database files that are created for temporary data storage. This data is then deleted after being processed further and stored in the output files. The details of such database files are not mentioned as these requirements arose during the development work and the details of these files are not important to the user. The TPS also creates several index files. The entire system handles just less than two hundred files and this includes all types of files required for both the modules, the simulation modelling system and the TPS.

5.4 TOOL PLANNING MODEL - THE PROGRAMMING LOGIC

The developed Tool Planning System (TPS) accomplishes the following objectives.

(1) Determine tooling requirements from the process plans ('WHAT' tools ?).

(2) Calculates required tooling quantity based on order batch size, machining requirements and tool life ('HOW MANY OF EACH' ?) for each component given in the production plan.

(3) Calculates the purchase order release time based on the tool required time, tool procurement lead time and loading time of parts on the shop floor ('WHEN' required ?).

(4) Aggregate the tooling requirements considering all the components in a product structure and putting them in appropriate time buckets.

This section describe how the data processing is carried out using the database files as specified in the earlier sections. A brief explanation on the programming logic of the TPS, together with its pseudo code is provided here.

There are two important routines in TPS. The first routine is called 'part tool calculations', which computes the tooling requirements on a part by part basis (the flowchart of the process is given in Fig 5.2). The second one is called 'week', which puts the requirements in appropriate time buckets based on planned order release date (plord_date) of components (explained with the help of a flowchart in Fig 5.3). This routine gives weekly requirements of tools, which is the final output of the TPS.

5.4.1 Part Tool Calculation Procedure *(ALL_TOOLCAL)*

It is assumed that MRP provides the parts that are to be produced in weekly time buckets. Each part has a unique part_no and is loaded on the shop floor by the planned order release date (plord_date). Each internal order is coded as order_no which represents the part_no and the order batch size of components (batch_size is dictated by the MRP). This information is stored in production plan (part_ord database).

At the start of each run, all the records in the temporary database are deleted. This is a separate routine. The data necessary to compute the requirements is retrieved from various database files (can be called as input files). These data attributes are temporarily stored in tool_req database. They are then processed further using equations [5.1] to [5.6]. These equations are given in the following pseudo code. The detailed listings in Foxpro are available from Appendix-1. The data processing logic is explained in Fig 5.2.

All the records from tool_req are transferred to all_req database after calculating the requirements for the first part (order_no). The records from tool_req are then deleted for storing the requirements for the subsequent order. However, the final output is stored in all_req database, where the data is accumulated for all parts that are listed in production plan (part_ord database). The all_req database is used as input for 'tool aggregation' process (refer to chapter 4 for more details).

```
BEGIN : PROCEDURE tool calculations
   USE part_ord database
   DO for all parts one by one
      RETRIEVE part_no, batch_size, plord_date (=reqd_date)
      LOCATE the process plan database for part_no
      USE the selected process plan database
      RETRIEVE opn_no, tool_code, machining_time
      STORE opn_no, tool_code, reqd_date in tool_req database
      USE tool_req database
      DO for all tools (tool_code) one by one
         RETRIEVE first tool (tool_code)
         LOCATE tool_code in tool_na database
         Equations [5.1], [5.2], [5.3], [5.4]
         IF tool is returnable, THEN
            Equation [5.5]
         ELSE (means if non_returnable, then)
            Equation [5.6]
         ENDIF
         STORE the calculated tool required quantity in all_req database
      ENDDO
   ENDDO
END :
```

$$\textit{total jobs} = \frac{\textit{unit life}}{\textit{machining time}} \qquad [5.1]$$

$$\textit{no. of tool changes} = \frac{\textit{batch size}}{\textit{total jobs}} \qquad [5.2]$$

$$\textit{no. of regrinds} = \frac{\textit{estimated life}}{\textit{unit life}} \qquad [5.3]$$

$$\textit{scrap quantity} = \frac{\textit{no. of tool changes}}{\textit{no. of regrinds}} \qquad [5.4]$$

$$\textit{total quantity}_{\textit{returnable}} = \textit{integer}\ (\textit{scrap quantity}) + 1 \qquad [5.5]$$

$$total\ quantity_{non-returnable} = integer\left(\frac{batch\ size * machining\ time}{estimated\ life}\right) + 1 \quad [5.6]$$

5.4.2 Weekly Requirements Procedure *(WEEK)*

The part tool calculation routine stores the requirements in all_req database on the basis of orders. Each order is released by the plord_date. This date is used to put all the requirements for that part in a weekly time bucket. For each new run of TPS, there is a new weekly requirement report that is generated as part of the TPS output. How this data is utilised for planning purpose is explained in chapter 7, where different tool planning strategies were developed using simulation. The logic of this routine is explained with the following pseudo code and the flow chart (Fig 5.3).

The tool planning module was developed in the above manner using various methodologies like DFDs, the Normalisation technique and the program design. The following chapter describes how the second module, the simulation modelling system was developed and integrated with this TPS.

```
BEGIN : PROCEDURE week
   USE part_ord database
   DO for all orders one by one (order_no)
      RETRIEVE order_no and planned order release date (plord_date)
      USE all_req database
      SCAN for selected order_no
         RETRIEVE tool_code (tcode) and required quantity (reqd_qty)
         USE week database
         CASE 1 - IF plord_date falls in week1, THEN
                  STORE tool_reqd_qty in week1
         CASE 2 - IF plord_date falls in week2, THEN
                  STORE tool_reqd_qty in week2
         CASE 3 - .........
         CASE 4 - ........
      ENDSCAN
   ENDDO
END :
```

Production Plan Database (PART_ORD.DBF)

Primary Key	Data Attributes
Order Number (order_no)	Part Number (part_no)
	Batch Size (batch_size)
	Planned Order Date (plord_date)

All Requirements Database (ALL_REQ.DBF)

Composite Key	Data Attributes
Order No. (order_no)	Workcenter No.(wcent_no)
Operation No. (opn_no)	Tool Code (tcode)
	Required Quantity (reqd_qty)
	Operation Time (run_time)

Tool Aggregate Database (TOOL_AGG.DBF)

Primary Key	Data Attributes
Tool Code (tcode)	Required Quantity (reqd_qty)

Part Number Database (P1000.DBF)

Primary Key	Data Attributes
Operation number (opn_no)	Operation name (opn_name)
	Workcenter number (wcent_no)
	Tool code (tcode)
	Set-up time (set_time)
	Machining time (run_time)

Tool Requirements Database (TOOL_REQ.DBF)

Primary Key	Data Attributes
Operation Number (opn_no)	Tool Code (tcode)
	Tool Required Date (reqd_date)
	Tool Required Quantity (reqd_qty)
	Estimated Tool Life (est_life)
	Operation Time (run_time)

Weekly Requirements Database (WEEK.DBF)

Primary Key	Data Attributes
Tool Code (tcode)	Week1
	Week2
	Week3
	Week4

Tool Characteristics Database (TCODE_NA.DBF)

Primary Key	Data Attributes
Tool Code (tcode)	Tool Name (tool_name)
	Returnable ? (returnable)
	Procurement Lead Time (proc_lt)
	Standard ? (standard)
	Class (class)
	Price (price)

Tool Life Database (TOOL_LIF.DBF)

Composite Key	Data Attributes
Tool Identification Code (tool_id)	Estimated Life (est_life)
Tool Code (tcode)	Unit Tool Life (unit_life)
	Available Life (ava_life)
	Number of Regrinding Events (no_regrind)

Tool Inventory Database (TOOL_INV.DBF)

Primary Key	Data Attributes
Tool Code (tcode)	Current Stock Level (in_stock)
	Minimum Stock (min_stock)
	Purchase order size (order_size)

Tool Engineering Database (TOOL_ENG.DBF)

Primary Key	Data Attributes
Tool Code (tcode)	Tool Size (tool_size)
	Tool Material (tool_matl)

FIG 5.1 RELATIONSHIP OF DATABASES

Fig 5.2 FLOW CHART OF TOOL CALCULATION PROCEDURE

/ljta rt procedure^
VALLJOOLCAL J

ARTJJRD \ PARTJiO INPUT i=i TO n
DATABASE --------> PART NO - i < -
FILE

PROCESS PART_NO LOCATE FOR
PLANNING PART TYPE
■DATABASE

OPERATION SEQ TOOL CODE

TOOLJEQ TOOL CODE
DATABASE

YES

CAL QTY WITH CAL QTY WITH
EQS 6.1 TO 6.5 EQUATION 6.6

TOOL CODE
ALLJEQ STORE THE TOOL
DATABASE REQUIRED QUANT
REQD_QTY

V

SELECT TFiE NEXT
PART, 1-1+i

END PROCEDURE
ALL_TOOLCAL

Fig 5.3 FLOWCHART OF WEEKLY REQUIREMENTS

/ START PROCEDURE'^
^WEEKLY REQS J

PARTJ3RD
DATABASE
.FILE

ORDER NO ' ^

/ INPUT)=1 TO n
ORDER NO = }

TOOLJEQ
DATABASE

TOOL REQD DATE

LOCATE FOR
PL.ORD.REL.DATE

ALLJEQ
DATABASE

ORDER NO (j)

LOCATE FOR
ORDER NO =>]

TOOL REQD DATE

PUT PLORD DATE
IN WEEK (1/2..1

TOOL CODE
REQD QTY
WEEK NQ,

WEEK
.DATABASE

SELECT THE NEXT
ORDER NO. 3-3+1

CHECK IF

YES

W___
END PROCEDURE "A
ALL_TOQLCAL J

SIMULATION MODELLING OF TOOL PLANNING SYSTEM

6.1 INTRODUCTION

Simulation has been used as a valuable tool for solving many complex manufacturing system problems. Having developed the tool planning system (TPS) for batch manufacturing, it is necessary to evaluate its effectiveness. The advantages and limitations of the TPS will not be fully understood, if the performance measurement parameters are not compared with the traditional tool stock control techniques.

With the help of simulation techniques, one can build a model of a dynamic production environment having a direct interaction with the tool management system. The model can be used for predicting the performance of the proposed TPS and can act as a guideline for developing new tool planning strategies. Such an exercise has been carried out here using a Database Management System and can be described schematically in Fig 6.1.

All the developed procedures can be grouped into three main categories, viz;

1. The core simulation engine.
2. The TPS routines to determine tool requirements and planning them in time

buckets.

3. The procedures to create the manufacturing environment which links the TPS with the simulation engine.

This chapter justifies the selection of a suitable simulation modelling system, explains the methodology behind the core simulation engine and illustrates how it is embedded in the selected database package. Furthermore, the simulation mechanism is explained along with the definition of its system parameters and assumptions made during the process of its construction.

6.2 SELECTING A SIMULATION SYSTEM

The selection of a suitable simulation system would depend upon the modelling requirements. Some of the prime requirements are given below,

(1) The system should be able to model the entities such as machines, parts, tools and a tool store with tool database.

(2) It is required that the simulation is of the discrete event type.

(3) The ability to treat tools as entities with attributes whose values change as the simulation clock advances.

(4) Ability to make changes in the database records of the selected database system as and when the tooling transactions take place through the tool stores.

(5) Ability to handle large amount of data.

Simulation models can be developed by using either a general purpose language (FORTRAN/PASCAL) or using simulation software [Carrie, 1988]. The drawback of using a general purpose language is that tremendous programming effort is required in developing models. This leads to time consuming tasks of error checking, validating and amending the model. Additionally, it would have been difficult to interface the models with a database system.

Another possibility was to make use of commercial simulation package. SIMAN-CINEMA has been the most popular package used for solving the manufacturing system problems. One of the advantages of using SIMAN would have been the display of graphical animation. However, the volume of data handled by the TPS would have been beyond its capacity. Furthermore, interfacing requirements with a database system would have not been resolved.

One of the potential candidates for modelling the tools has been the GASP methodology [Pritsker, 1974]. This is a generic method which was developed in FORTRAN language. It was traditionally implemented in FORTRAN, but the concept can be translated into other languages. It is a discrete event simulation method and could provide the facility of creating and handling tools as the entities (in addition to the parts) and execute the events when the system reaches the event time [Perera, 1988].

It is not feasible to interface Foxpro with FORTRAN based routines of GASP. However, it was possible to code the GASP routines into Foxpro without much difficulty. By coding the entire system in Foxpro, the integrity of the model (the TPS and the simulation engine) could be maintained. One of the distinct benifits of using GASP methodology was that it could handle large amounts of data by creating arrays. Additionally, this gave an opportunity to understand the mechanism of simulation, which in turn means a better control over the entire process of modelling, fine tuning and error checking procedures. This needed less intensive labour in verifying the successful operation and development of further simulation models. Thus, the choice of incorporating the GASP methodology into Foxpro for modelling the TPS was made for this work.

However, the selected approach did not allow opportunities to have a graphical animation in Foxpro. This means that the entire mechanism of the simulation model had to be examined on the basis of the numerical and graphical data alone. Lack of visual display of the model has led to this difficulty, which was anticipated.

6.3 DISCRETE EVENT SIMULATION

The model developed consists of a job shop scenario with workcenters, parts and a tool store. It can be represented in Fig 6.2 (the details on the model definition and assumptions made can be made available from section 6.6). The parts are processed on different workcenters and they flow according to the process plans. When the part arrives, the required tool as specified in the process plan, is issued by the tool store. Each process consumes a certain amount of tool life. Such a manufacturing scenario is regarded as a discrete type of simulation. A typical simulation model is represented by the ENTITIES, ATTRIBUTES, EVENTS, QUEUES, ACTIVITIES and STATES [Carrie, 1988]. These are explained below.

ENTITIES - The 'parts' and 'ordered tools' form the entities in this model. There is also a dummy entity used for initialising and advancing the simulation clock.

ATTRIBUTES - Each of the above mentioned entities have their own attributes. The values of these attributes change as the entities flow through the system during the simulation. For e.g. the entity 'part' has the following ten attributes associated with it.

1. Event time
2. Event code
3. Part number
4. Batch size
5. Operation number
6. Workcenter number
7. Operation time
8. Tool code

9. Tool required quantity

10. Order number

EVENTS - Event occurs when a certain activity begins or ends. Each entity has 'event time'(attribute 1) and 'event code' (attribute 2). The 'event code' represents an activity that needs to be carried out when the system clock reaches the 'event time' (attribute 1). The entities are chronologically queued in the 'event queue' and are executed in increasing order of the 'event time'. Some of the important events handled by the system are described below. The event occurs when,

(a) the part is loaded on the workcenter.

(b) the part is unloaded from the workcenter at the completion of the operation.

(c) the purchase requisitions are sent for tools (or when the tools are ordered).

(d) the tools are received by the tool stores.

(e) the part begins to wait in the 'waiting queue' due to tool shortages.

(f) the part ends its waiting process and is loaded on the workcenter.

There are other events such as checking the tool stock levels. If the tool stock levels are below the minimum required, then those tools are purchased and restocked. These events take place according to the tool restocking rules. Each model has a unique tool replenishment strategy, leading to different nature of events being built for each model. These strategies are outlined in detail in Chapter 7.

The 'event code' assists in recording many time dependent variables. For e.g. the production delays ('total_wait') can be calculated by taking the difference between the 'start' and 'wait' times.

QUEUES - The simulation engine consists of a queue called as the 'event' queue and the other two types are 'arrive' and 'wait' queue. 'Event' queue has all the entities of the system arranged chronologically, irrespective of the nature of the entity. This process of arranging the entities in a sequence is carried out by the simulation engine. Each workcenter in the model has one 'arrive' and one 'wait' queue. When the part arrives at the workcenter , it is put in the 'arrive' queue. The tools required for that operation are then issued from the tool stores. If there are any tool shortages, then the part is transferred to the 'wait' queue and is held until the tools become available. The 'event' queue may consist of any entities (parts or tools on order), whereas, the queues at the workcenter contain only parts as entities.

ACTIVITIES - Activities are the processes such as the part being machined at the workcenter or the tools being procured. Every activity has a certain length of time which is taken into account while calculating the activity finish time. This is handled by the engine routines. For e.g. the tool procurement time is added to the time at which the order is placed and the tools are received when the system clock attains the resultant time.

STATES - Each entity has a 'state', for e.g. when the parts are loaded, the machine is set to 'busy' and at the completion of the operation, it is set to 'idle'. Similarly, when the order is placed for procurement of certain type of tool, then the entity 'tool on order' has the flag 'ordered ?' which is set to 'yes'. As soon as the tools are received, the flag is set to 'no'. With this facility, the activities can be logically controlled.

6.4 THE MODELLING PROCEDURES

The simulation engine was linked to the TPS procedures within Foxpro. The original routines of GASP were very comprehensive and versatile. The job shop model did not require all these routines and therefore a cut down version of GASP was used. The necessary amendments were made to serve the purpose of modelling. This in effect increased the speed of the simulation process. The following section explains the mechanism of simulation models on the basis of the coded routines. This maintained the consistency in explanation method, which can make the reader understand this matter better.

6.4.1 GASP Procedures

The selected GASP routines include the ENGINE, REMOVE, QUEUE, EVENT and INITIALISE. There were other subroutines such as VARIABLES, ENVIRONMENT, DISNSET and QDATA, all written in Foxpro to fulfil the modelling requirements. These routines form a part of the simulation engine and are stored in separate file of

GASP procedures, called as 'gasp.prg' (details can be made available from Appendix-2). The names of the GASP variables have been unchanged while coding in Foxpro, so that the programs could be verified by referring to the original GASP routines. The following section gives a brief description of their functions.

ENGINE This is the simulation executive, and activates the events in a chronological order until the system reaches the end of simulation period. All models have total simulation period of eleven weeks. The system clock is advanced by this routine and hence, this forms an important routine to control the simulation.

REMOVE When called, this routine pulls out the entity from the desired queue.

VARIABLES All the global variables required to run the gasp routines are defined here. There are many variables which are represented by arrays of numbers. The size of these arrays are specified in this routine.

QUEUE This routine puts the desired entity into the required queue. It then rearranges the sequence of the entities in the increasing order of their 'event time'. A first-in-first-out (FIFO) logic has been used.

EVENT This routine has all the activities that are needed for simulation. These activities are defined by the 'event code'. These events are triggered when the system clock reaches the 'event time'. ('Event' is also elaborated in section 6.3.)

INITIALISE This routine initiates the array pointers.

6.4.2 Procedures for linking Simulation Engine with TPS

There are other routines written for creating a manufacturing environment. These routines act as a bridge linking the TPS and the engine, and are explained below. (The details can be made available from appendix-1).

ENVIRONMENT This routine set the programming environment that is necessary for running the system.

CLEANDATA Prior to every simulation run, certain records from the database files are amended depending upon the initial conditions of the model desired. The records from many temporary database files are also deleted for storing new data.

INITIALISE The arrays required for simulation are created and initialised using this routine.

SIM_VARIABLES This routine is created to define the variables which are required to run the job shop model. For e.g. the maximum number of machines allowed in the shop can be defined here.

GETATRIBS This routine extracts the required information from various database files and defines them as attributes of the entities. The random function is used to generate randomness in some attributes. Finally, the entities 'parts' are created by inserting these attribute values into the attribute array and putting them in the 'event queue' for loading.

START_OP The parts from the 'arrival queue' are loaded on the workcenter. The tools for the operation are checked for its availability, if available then the parts are processed, otherwise, they are put in the 'wait queue' until the tools become available.

WAIT_OP This routine is similar to the 'start_op', except that the parts in the 'wait queue' are considered instead of 'arrival queue'.

END_OP This routine is executed at the completion of operation. Again certain attributes of the entity are amended during the execution.

6.5 SIMULATION MECHANISM

The library of procedures are stored in either the 'gasp.prg' (as explained in section 6.4.1) or the 'model*.prg' type files. Each model has a unique model*.prg file, but the 'gasp.prg' (program file) remains unchanged and acts as a supporting simulation engine for all model type files. The model type files contain both, the TPS procedures (as explained in chapter 6) and other routines (as explained in section 6.4.2) for creating a job shop scenario. The models developed are grouped into two categories.

Type A : The models of job shop with tool stores that operates without using the outputs produced by the proposed TPS. Instead, it uses the traditional stock control techniques, such as 'fixed order-flexible time' and 'fixed order-fixed time'. Details on various tool replenishment strategies can be made available in Chapter 8. There are three models developed with three unique strategies.

Type B : There is only one model being developed with similar manufacturing environment as above, but having tool stores operating on the basis of TPS.

The purpose of developing the 'A' model (without the TPS) was to compare the effectiveness of performance of TPS with the traditional approaches. By building more than one 'A' models, the TPS performance can be compared with not just one rule but several other strategies. This gives an opportunity to substantiate the argument with extra confidence.

Although there are mainly two types of models, the program routines are very similar to one another, except for a few differences. For example in order to operate the tool stocking activity on the traditional stock control rules, an additional routine had to be written for 'type A' models. All other features are kept unchanged, so that the models could be compared under identical experimental conditions. The mechanisms of both the model types, i.e. with the TPS and without the TPS are diagrammatically represented in Figs 6.3 and 6.4 respectively.

The initial procedures (ENVIRONMENT, VARIABLES, INITIALISE, SIM_VARIABLES) are activated when the model begins to run. These procedures create and define the required manufacturing environment. The model then executes the TPS procedures (ALL_TOOLCAL, WEEK_REQ as illustrated in chapter 6). The most important output of this process being the tool requirements in weekly time buckets. With this, the model has now a fictitious job shop and tool stores. It is then ready to commence the simulation of part loading and part machining.

As explained in Section 6.4, GETATRIB creates entities and schedules them in 'event' queue on the basis of 'event time'. The entire control of event execution process is then delivered to ENGINE. The simulation runs until ENGINE registers the end of the simulation period.

In the end, the model executes the procedure PRINT{A*/B*} (depending upon type 'A' or type 'B' model), which is written for collecting, further processing and printing the output data in the desired format for analysis.

6.6 MODEL DEFINITION AND ASSUMPTIONS

The entire model is shown in Fig 6.2 and can be defined with the parameters which are explained in this section. A typical batch manufacturing situation with different machines, queues and tool stores is being modelled. There are two configurations of the job shop model that were used in this study. Both the configuartions have similar features and use exactly the same data generated by the modelling system. The difference between these configuartions are the number of workcenters and the variety of tools. The second configuration is larger than the first. The purpose was to study whether the TPS is effective in large size production environment. In the first configuartion (configuration #1), five workcenters were used with eleven tool types, whereas the second configuration (configuration #2) has ten machines and sixteen type of tools. The part variety is therefore doubled to eight in the second configuration. The production orders are also increased from thirty to thirty five. The total simulation period has been unchanged in both the configurations. The following section gives the definition of both the configurations.

List of model parameters :-

- Number of workcenters = 5 in config #1 and 10 in config #2. These workcenters are assumed to have predominantly metal cutting activities.
- Number of part types = 4 in config #1 and 8in config #2.
- Number of production orders over the period of simulation = 30 in config #1 and 35 in config #2
- The production batch size of parts vary from 200 to 2000
- Total simulation period = 11 weeks (each week = 50,000 time units) for both the configurations.
- Number of event queues = 1
- Number of 'arrive' queue = one for each workcenter
- Number of 'wait' queue = one for each workcenter

Tooling information :-

- Number of tool stores = 1
- Number of tool variety = 11 for config #1 and 16 for config #2 (only cutting tools are considered and includes both returnable and non-returnable types)

During the process of model building, various assumptions were made. These are also discussed in this section.

Assumptions :-

All the parts that are planned for production (as specified in the part_ord database file) are loaded within seven weeks from the start of the simulation. This period was selected so that the system could be filled with the entities and the data could be generated for output analysis, starting from 'week-two' until 'week-six' (truncating the effect of initial conditions).

A dummy entity is used for both initialisation as well as for advancing the system clock. The system clock is increased by 2000 time units (which is approximately two hours). This denomination was selected so that the averages of certain parameters (e.g. tool inventory level) over a period could be calculated. The time units of all time dependent variables are relative to the simulation clock units.

It is assumed that there is only one vendor that supplies the necessary tooling. The production system is not given a choice to select a suitable vendor from the cost and service point of view. In practice, there is always a choice of tool vendors but the problem is kept simple as this is not the focus of attention.

All the models assume that tool inspection activities are carried out before restocking the new and used tools. There is no time delay added for these activities and these times are assumed to be insignificant compared to the process times of jobs.

In the entire exercise, only the single tools are considered. There are multiple tooling set ups required in certain operations. In these cases, the process does not begin until all the tools required in the set up become available. In the case of multiple tooling, which is made up of different components, additional activities like kitting and assembly need to be carried out. Therefore, planning and control of such type of tools is more complex than even returnable types. These tools are not accounted in the simulation. However, these tools are treated as single tools in this exercise. Further research along these lines is recommended with the advent of multiple tooling.

There are no job priority rules being considered in the models. Each set of job priority rule will result into a unique set of simulation results. Which means that there will not be any variable on which a comparison between the model's performance can be made. It is anticipated that the proposed TPS will take into account the tool requirements of the job being loaded on the workcenter. However, the job priority rules will have significant effect on the tool availability, if the traditional stock control models are used instead of the TPS. The First-in-first-out logic has been used in all the models. The simulation study of tool availability using a single machine with job priority rules has been carried out by Melnyk[1]. Further work is recommended to study the effect of job priority rules on tool availability within the job shop model with more than one workcenter.

The job ahead of the 'start queue' is selected and the tool availability is checked. If the tool is available, then, the job is processed immediately, otherwise, it is put in the 'wait queue'. If the job goes to the 'wait queue', then the next job in the 'start queue' is selected and the required tool is checked in the similar manner. The machine is not allowed to remain idle unless all the jobs in the 'start queue' face shortages.

If the job priority rules had been given in the models, then the problem in the simulation model would have been complex and beyond the scope of the objectives of the study. However, advance versions of these models having this facility could be developed for further study.

Random Number Generation :-

Foxpro provides a function called IRAND() to generate randomness in the system parameters. It was assumed that these parameters will have uniform random distribution. These are listed below,

(a) part loading / release time

(b) batch size of production orders

(c) operation times (It was assumed that the machine set up time for tool changes would be small compared to the operation times of the batches and therefore it was included in the operation time).

6.7 MODEL VERIFICATION AND VALIDATION

Computer simulation extends well beyond the task of mere programming and model building. Many decision makers regard this as an exercise of computer programming, but it is equally important to verify and validate the models and use appropriate statistical technique for analysis of output data. Computerized model verification is a process of ensuring that the computer programming and the implementation is correct in accordance with the conceptual model [Sargent,1992].

All the developed models were tested individually to confirm the correctness of their operation. During the testing procedures, the individual program routines within the model were verified with valid input data (this is also called 'module test' in software engineering terms). The output data was tallied with the manual calculations.

In order to verify the correctness of the simulation logic, manual calculations have to be performed. This involves ensuring that the entities such as jobs are released on the shop floor at the required time. This can be known from the information such as planned order release date (the production plan database). This verification is carried out before incorporating the randomness in the variables. In addition to the time when the entities are released, it is also important to ensure the sequence in which they are released. As the simulation progress, the values of the entity attributes (such as the machine number and the operation number) change. Therefore, it is necessary to monitor such changes for verification. The details on the entity attributes can be either printed or viewed on the screen during the simulation.

Once the individual programs were proven to be error free, then all these programs were integrated and the entire model was tested (this is called 'link test'). Thus, all the models were verified using this approach.

The model validation involves determining that the model's output behaviour has the accuracy required for the model's intended purpose [Sargent, 1992]. Sample of the production plan as an input to the simulation model was tested over a period (eight weeks in our case). A typical output of weekly tool requirements generated by the TPS is shown in Table 6.1. The output at the end of fixed time interval (state of the model at the end of every week) was recorded and printed. This data was then studied to evaluate the model's behaviour over this period. Some of the tasks in this exercise included analysing the changes in entity attributes during the simulation, studying the engine's event execution mechanism and monitoring the system's response to different events. One of the most critical events from the study objective was when the system registers production stoppage due to tool shortage.

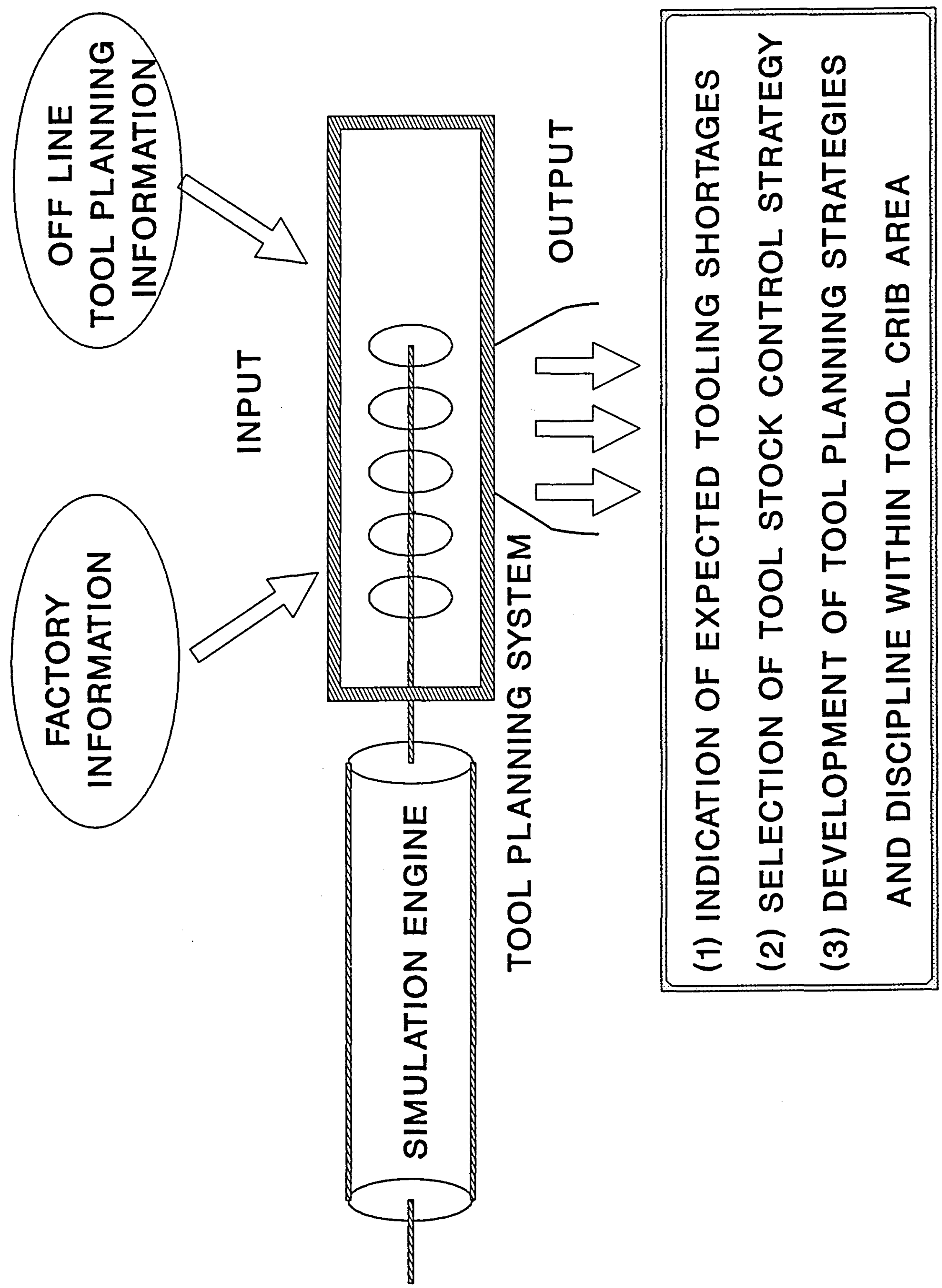
In some cases, additional assumptions had to be made to maintain the consistency in the data format of the system variables. Several simulation runs had to be carried out to study the exact behaviour of the models before commencing the experimentation for actual results.

The following chapter describes all the models and the associated tool planning strategies in detail. The simulation results are analyzed and discussed for assessing the effectiveness of TPS.

Table 6.1 Weekly Requirements Report

Tool No (tcode)	Week Number							
	1	2	3	4	5	6	7	8
10	0	0	0	0	0	0	0	0
11	33	10	11	5	6	5	0	5
12	9	2	2	0	0	3	0	0
13	5	3	3	2	1	2	1	1
14	3	2	2	3	3	2	0	3
15	1	2	1	2	1	2	2	0
16	1	2	1	2	2	2	1	1
17	0	1	0	2	2	0	3	0
18	4	4	2	6	1	6	4	0
19	24	5	6	0	0	0	0	0

FIG 6.1 INTEGRATED TOOL PLANNING SYSTEM



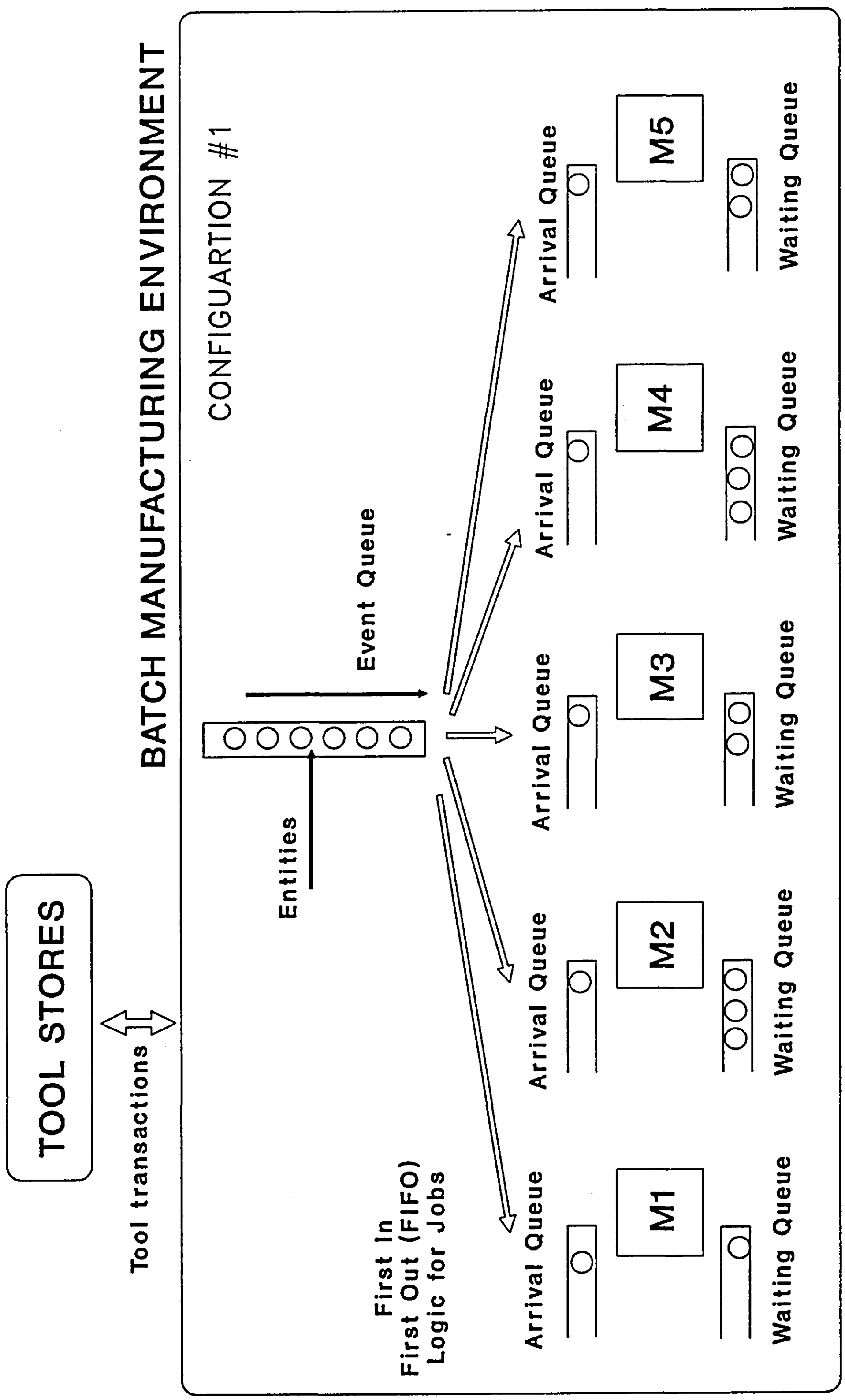


Fig 6.2 Simulation Model of Batch Manufacturing

FIG 6.3 MECHANISM OF MODEL WITH TPS

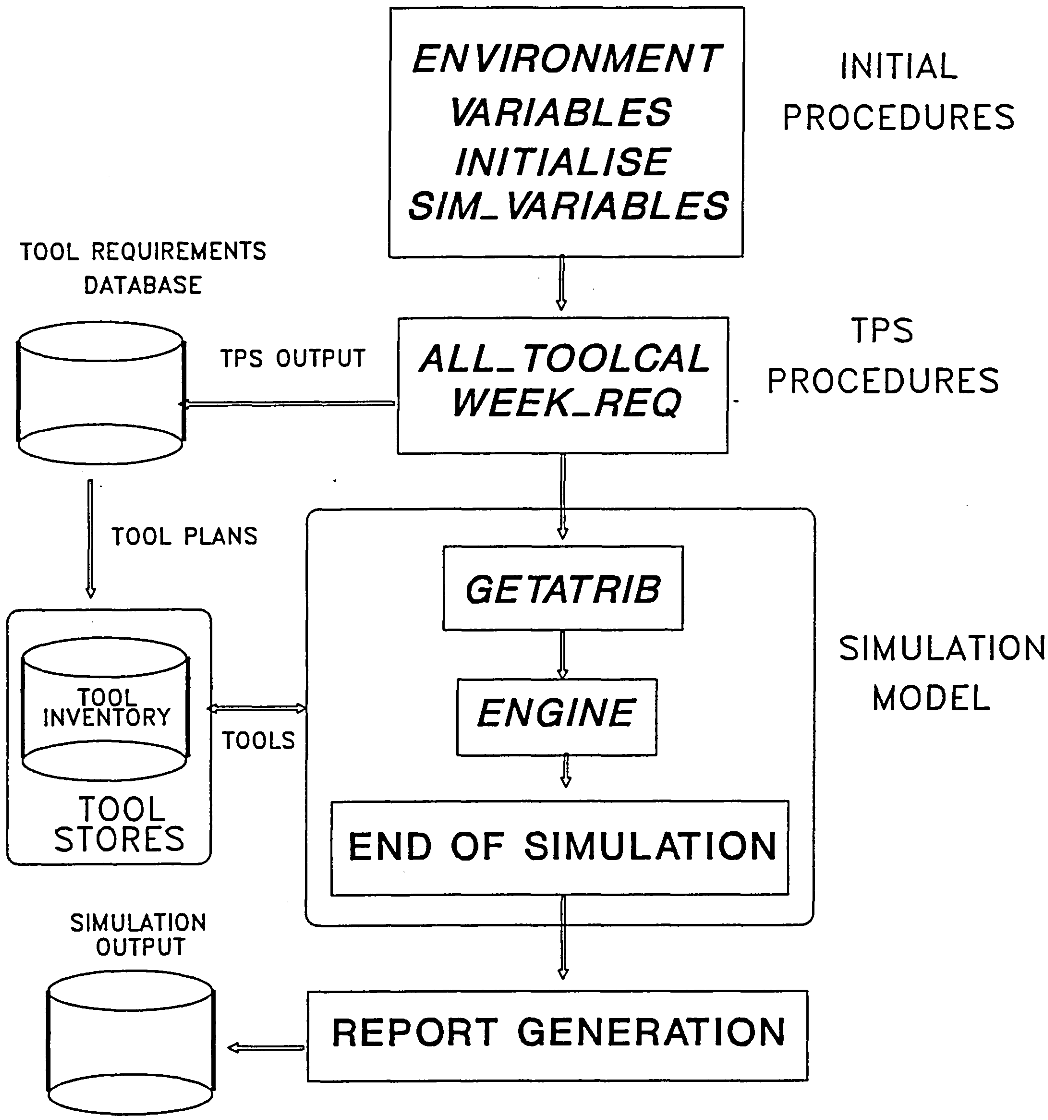
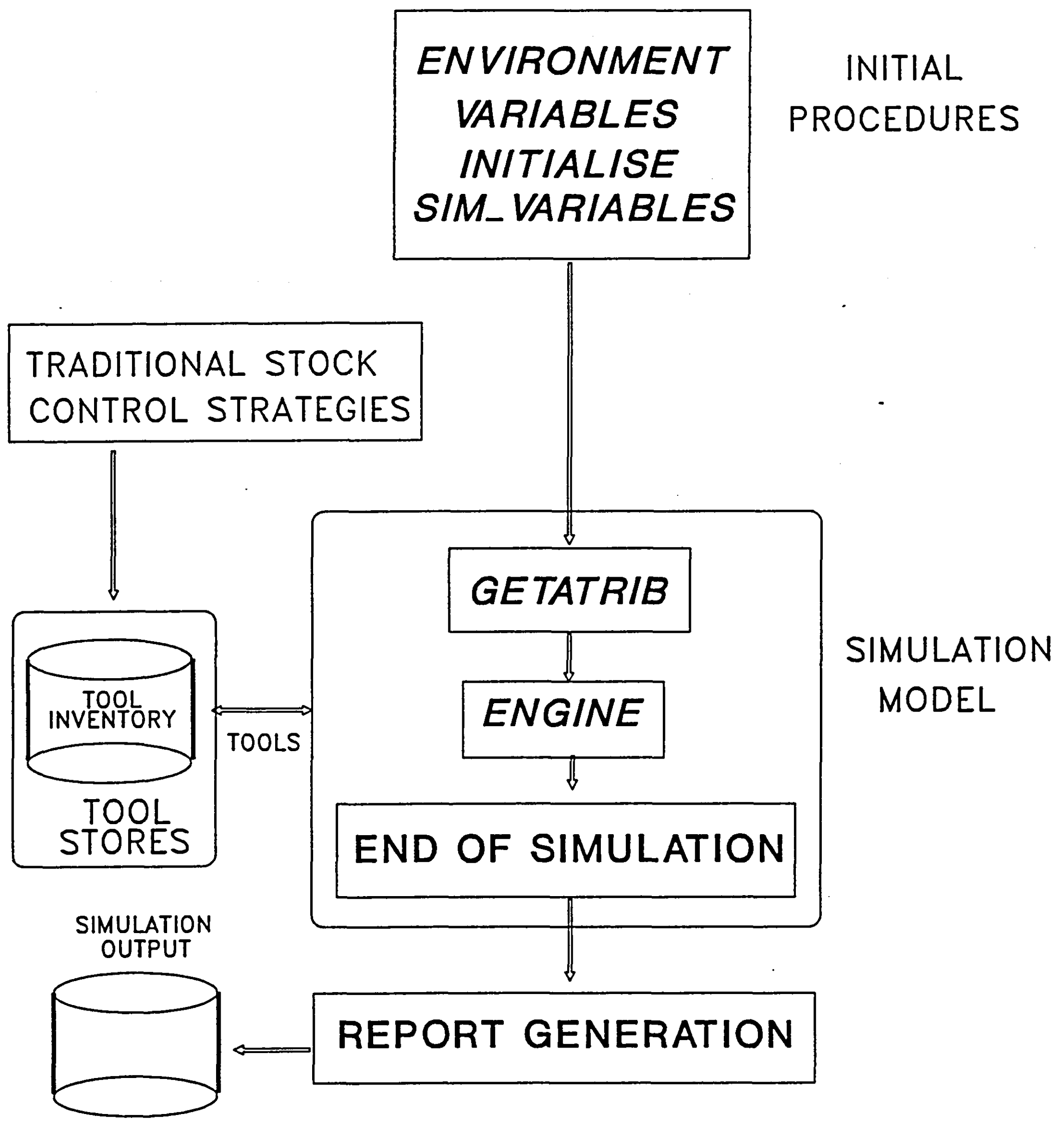


FIG 6.4 MECHANISM OF MODEL WITHOUT TPS



7

EVALUATION OF TOOL PLANNING SYSTEM

7.1 INTRODUCTION

The major objective of the simulation modelling is to determine the effectiveness of the proposed Tool Planning System (TPS) and compare it with the traditional tool control approaches. Each of these stock control approaches resulted into a different model having unique characteristics. These developed models are explained in detail.

Certain system parameters were identified to measure the performance of these models. The selection of a suitable statistical method is justified and is illustrated in later sections of this chapter. In the end, the experimental results are discussed with the view of evaluating the developed TPS.

7.2 SIMULATION MODELS AND RELATED TOOL STOCKING RULES

The effectiveness of each of the models can be compared with one another by choosing the appropriate performance measurement parameters. The most commonly used model parameters such as machine utilization and production rate can not be used to compare the effect of the developed TPS and the traditional tool stocking rules on the performance of a production system. Therefore, alternative parameters had to be identified.

The maximum benefits (from the viewpoint of tooling) in any production system can be made if there are minimal production stoppages with optimum inventory levels. It is also now well established that significant costs are incurred in 'hot purchases' of tools [Kravitt, 1988]. This is a potential cost saving area where appropriate tool planning technique can play a vital role. These two criterion were selected as the basis for comparing the performance of the developed models. It is also important to observe the effect of different planning rules on the tool inventory levels. Therefore, the important performance measurement parameters for comparison are the *'Number of tooling shortages'*, *'Number of tooling purchases'* and the *'tool inventory level'*. Additional program routines had to be written to collect the output data on these parameters. A detailed discussion on the simulation results is given in section 7.6.

As explained in chapter 6, section 6.5, there are two main categories of models built for this exercise. Each of these models operate on a unique tool replenishment rule. There are four different models, of which three are based on traditional tool replenishment approaches (designated as A1, A2 and A3). The fourth model (B1) is designed to operate on the information supplied by the Tool Planning System (TPS). The 'A' type models are based on some of the inventory control rules proposed by Long[1] [1991]. Long suggests several tool replenishment rules such as Fixed Order Point (based on minimum/maximum levels) and Flexible Order Point. The model 'A's do not have any link with the TPS. The following section describes the operating rules of these models in detail.

7.2.1 Type 'A' Models (Without the TPS)

Model-A1 : Fixed Time, Flexible Number of Purchases (FIX-T, FLEX-P)

The inventory level of all tools is checked at the end of every week. The minimum level of inventory is predefined on ad-hoc basis. The initial inventory at the beginning of simulation is equal to the minimum stock level. Only those tools whose inventory drops below the minimum level are procured. The quantity ordered is also set on ad-hoc basis. This quantity is constant and is termed as 'order size'. Each tool type has a different 'order size', but this quantity remains constant for that tool throughout the simulation run. All the 'A' models have the same 'order size'.

Each tool type has a different procurement time and is added to the time when the tool is ordered. Therefore, tools are received at different time points depending upon thHxr individual procurement time.

Model-A2 : Flexible Time, Flexible Number of Purchases (FLEX-T, FLEX-P)

The frequency of tool stock checking activity is changed from weekly (as in the above model) to virtually an hourly basis in this model. This makes the model dynamically responsive to changes in inventory levels. This strategy is incorporated in this model.

At the end of every hour, the stock levels of all tools are checked. When the level drops below its minimum required, the system registers the replenishment event and immediate action is taken to procure that type of tool. The tools are received at different time points similar to Model-A1. Again, only those tools with inventory below minimum are purchased.

Model-A3 : Fixed Time, Fixed Number of Purchases (FIX-T, FIX-P)

In this model all the tools are replenished at a regular time interval of one week regardless of the stock level. Every tool type has a different 'order size' but this quantity is constant at each time the tool is bought. The orders for procurement are placed in such a way that all tools are received at the beginning of each week. The initial inventory is equal to the minimum stock level as in all the above 'A' type models.

7.2.2 Type 'B' Model (With the TPS)

Model-B1 :

This model runs on the tool replenishment rules as specified by the TPS. The tool requirements are determined by the TPS and are provided to the simulation model. The requirements are stipulated on a weekly basis by the TPS (the time bucket can be changed to a 'daily' or 'monthly' basis depending upon the individual requirements, more information can be made available from Chapter 6). Therefore, all the procurement activities are geared towards the required date and scheduled backwards taking into account their individual procurement time. The tools are received and restocked at the beginning of the week in which its tool required date fall under. The order size of individual tool type varies according to the tool requirements plan of TPS. The initial inventory level also depends on the requirements of the first week.

The following section explains the statistical method used for analysing the identified performance measurement parameters of each of the above models.

7.3 ANALYSIS OF RESULTS

In order to make the comparison between various models, they should operate under identical experimental conditions with the same input data. The main input to the models was the production plans, process plans and tool master details, the consistency of which is maintained in all the models.

In a simulation experiment, the input model parameters (such as part arrival time, tool procurement time) are the randomly generated values. Therefore, a single simulation run is not sufficient to estimate or to compare the output parameter with other models. Hence, a number of simulation runs need to be carried out to obtain sufficient sample data. An appropriate use of statistical technique is essential before the decisions regarding the selection of the 'best' or the 'second best' model can be made.

Since on many occasions, the simulation output data are dependent, a classical statistical analysis method based on Independent Identical Distribution (IID) observations is not directly applicable. There is no perfect solution available to obtain the accurate estimates of the model's true parameters.

However, Law and Kelton [1982] have suggested few methods for analysis, based on the nature of the simulation experiment. They classified simulation types into two categories, a 'terminating' simulation and a 'steady state' simulation.

> A *terminating simulation* is one for which the desired measures of performance are defined relative to the interval of simulated time $[0,T_E]$, where T_E is the instant in the simulation when a specified event E occurs.

A *steady state simulation* is one for which the measures of performance are defined as limits as the length of the simulation goes to infinity.

In our case, the simulation terminates at the end of the eleventh week. The output data is collected from the beginning of the third week until the end of the seventh week (so, the total period of four weeks is used for output data analysis). (The reasons for truncating the initial and final conditions of the run are given in section 7.5.1). This is a terminating type of simulation. The following section gives a brief description of analysis methods available for terminating type simulation and gives reasons for selecting an appropriate one.

7.4 SELECTING A SUITABLE STATISTICAL METHOD FOR ANALYSIS

A method of averages have been used most commonly for comparing two or more systems. For example, for a given model, the average number of shortages (s_r) over the period of steady state for simulation runs from r=1 to n, can be defined as,

$$Average\ No.\ of\ Shortages = \frac{\sum_{i=1}^{i=n} s_i}{No.\ of\ Simulation\ Runs\ (n)} \qquad [7.1]$$

However, it may result in misleading conclusions [Law and Kelton, 1982]. A method of proportions is also available. In this method, the proportion of shortages in the given time interval, say $I_i[a,b]$, where i=1 to N (where N=number of intervals) are determined by the equation,

$$\textit{Estimate of No. of Shortages} = \frac{\sum_{i=1}^{i=N} I_i[a,b]}{N} \qquad [7.2]$$

Proportions are just the averages of legitimate random variables and therefore the results obtained are not sufficient for comparing two systems. Thus, alternative methods need to be adopted for analysis.

It is important to estimate the parameter close to its true value in order to compare it with output of the other models. There are two well known methods available for terminating type of simulation, 'Batched Means' and 'Independent Replications'. Although, the method of batched means has been a commonly used technique, Law [1977] claims that there is a possibility of correlation among the batched means, which seems to be the most deleterious. (Despite the assumption that the batched means are approximately IID random variables with unknown mean and variance).

Therefore, for the above reasons, the method of independent replications has been adopted in this exercise. Sample means of replicate means are taken for analysis and comparison of the performance measurement parameter.

In order to avoid poor estimates of random variables, care is taken to avoid the initialisation bias by truncating the output data (data truncation is explained in section 7.5.1). It is also ensured that the replicate means are independent between replications but not necessarily within a replication. This is achieved by initializing the simulation with different random seed for each replication.

There is evidence available that the method of independent replications has been found to be implemented successfully for the selection problem [Law & Kelton, 1982]. This method is extended to calculate the weighted sample means which is finally used to compare the output parameter. There were three important parameters identified for comparison, viz, the inventory levels, the number of purchases and the number of shortages. The selected method is used for each of these parameters. The entire method is explained in the following section.

7.5 METHOD FOR MEASURING SYSTEM PERFORMANCE PARAMETER

Let x_1 be the random variable of interest (for comparing the performance) from the 'r'th replication of the 'i'th model. There are four models under consideration, therefore i=4. All x_1, x_2,..x_m are assumed to be the Independent Identical Distributions (IID) random variables, and the runs for each model are independent of each other.

Replication 1 : x_1, x_2,...x_m $\Rightarrow$ mean X_{i1}

Replication 2 : x_1, x_2,...x_m $\Rightarrow$ mean X_{i2}

:   :

Replication r : $x_1, x_2, \ldots x_m \Rightarrow$ mean X_{ir}

$$\overline{X}_{i1}=\frac{x_1+x_2+..+x_m}{m} \qquad [7.3]$$

for the first mean and similarly,

$$\overline{X}_{ir}=\frac{x_1+x_2+..+x_m}{m} \qquad [7.4]$$

for the rth mean. The size m *(m=3 is the number of observations within a replication)*, is selected small enough, because as the value of m increases, the method of batched means become approximately equivalent to the method of independent replications.

Let $\mu_i = E(X_{ir})$ be the sample mean of the replicated means, which can be calculated as follows,

$$\mu_i = E[\overline{X}_{ir}] = \frac{\overline{X}_{i1}+\overline{X}_{i2}+\ldots+\overline{X}_{ir}}{r} \qquad [7.5]$$

Let $\{\mu_i\}_l$ be the 'l'th smallest of the μ_i's, so that,

$$\mu_{i_1} \leq \mu_{i_2} \ldots \leq \mu_{i_k} \qquad [7.6]$$

The goal is to choose the system with the smallest μ_i's from equation 7.6. The inherent randomness of the observed x_m's implies that one can never be absolutely sure that one makes a Correct Selection (CS), but one could prespecify the probability of CS.

The problem of selection can be formulated in a following manner. We want the probability $P\{CS\}$ such that, $P\{CS\} \geq P^*$ provided that $\{\mu_i\}_2 - \{\mu_i\}_1 \geq d^*$, where the minimal CS probability $P^* > 1/k$ and the 'indifference' amount $d^* > 0$. 'k' represents the number of systems under consideration. The values of P^* and d^* need to be specified. It was decided to argue our case with at least 90% probability ($P^* = 0.90$) and $d^* = 1$.

This statistical method involves two stage sampling for each model. In the first stage, a fixed number of replications of each model are made. Then the resulting variance estimates are used to determine the additional replications required for each model in order to reach a decision.

Let n_0 be the number of replications of each of the models in the first stage of sampling (where $n_0 \geq 2$). It is recommended that n_0 be at least 15, otherwise, there is a possibility of getting poor estimate of $s_i^2(n_0)$, which could lead to an unnecessarily large value of N_i. Therefore $n_o = 20$ was selected for this problem. The sample means (equation 7.7) and variances (equation 7.8) as given by Law and Kelton [1982] are,

$$\overline{X}_i^{(1)}(n_0) = \frac{\sum_{r=1}^{n_0} X_{ir}}{n_0} \qquad [7.7]$$

for $i = 1, 2.. k$, (where $k=4$ in our case). The total sample size N_i needed for the model i can be calculated with equation 7.9.

$$s_i^2(n_0) = \frac{\sum_{r=1}^{n_0} [X_{ir} - \overline{X}_i^{(1)}(n_0)]^2}{n_0 - 1} \qquad [7.8]$$

$$N_i = \max\left\{ n_0 + 1, \left\lceil \frac{h_1^2(s_i^2(n_0))}{(d^*)^2} \right\rceil \right\} \qquad [7.9]$$

The value of h_1 depends on k, P^* and n_o and can be found in Law and Kelton [1982, page 329, Table 9.7, Appendix 9A]. The value of d^* can be chosen as either 1 or 2. The greater the value of d^*, the lesser the number of additional replications needed for the second stage sample means (d=1 has been selected in this study). Having calculated $(N_i\text{-}n_0)$ more replications of model i for second stage, the second stage sample means are determined from equation 7.10.

$$\overline{X}_i^{(2)}(N_i - n_0) = \frac{\sum_{j=n_0+1}^{N_i} X_{ij}}{N_i - n_0} \qquad [7.10]$$

The weight W_{i1} is calculated from equation 7.11. (The definition of W_{i1} can be made available from Law and Kelton [1982]).

$$W_{i1} = \frac{n_0}{N_i}\left(1 + \left\{ 1 - \frac{N_i}{n_0}\left[1 - \frac{(N_i - n_0)(d_*)^2}{h_1^2(s_i^2(n_0))} \right] \right\}^{1/2} \right) \qquad [7.11]$$

and $W_{i2}=1 - W_{i1}$, for $i= 1, 2 .. k$. The weighted sample means are defined with

$$\tilde{X}_i(N_i) = W_{i1}\overline{X}_i^{(1)}(n_0) + W_{i2}\overline{X}_i^{(2)}(N_i - n_0) \qquad [7.12]$$

Thus, the model with the smallest value of the weighted sample can be selected as the best of the k options with confidence P^*.

SAMPLE CALCULATIONS

Say for example, the number of shortages for Model-A2 is being assessed. The sample mean of the replicated mean from the observations is $X_{2r}=\mu_2=4.36$ (calculated from equations 7.4 and 7.5).

In the first stage sampling, 20 observations were taken. Therefore, $n_0=20$ and the sample means calculated from equation 7.7 can be represented as,

$$\overline{X}_{ModelA2}^{(1stStage)}(20) = 4.36 \qquad [7.13]$$

The variance calculated from equation 7.8 is

$$s_{ModelA2}^2(20) = 1.08 \qquad [7.14]$$

The total sample size N_i is calculated from equation 7.9. The values $h_1=2.34$ and $d=1$ for $P^*=0.90$ are taken from Law and Kelton [1982]. This could be represented as,

$$N_{ModelA2} = \max\left\{(20+1), \left\lceil \frac{(2.34)^2\ (1.08)}{(1)^2} \right\rceil\right\} \qquad [7.15]$$

$$N_{ModelA2} = \max\ \{\ 21,\ 5.91\ \} = 21 \qquad [7.16]$$

The number of replications required in the second stage sampling is $(N_i - n_0) = (21-20) = 1$. The second stage sample means are calculated from equation 7.10, which is shown in the following manner,

$$\bar{X}^{(2nd\ Stage)}_{ModelA2}(1) = 3.66 \qquad [7.17]$$

The weights as calculated from equation 7.11 and can be represented as,

$$W_{(ModelA2)1} = \frac{20}{21}\left(1 + \left\{1 - \frac{21}{20}\left[1 - \frac{(21-20)(1)^2}{(2.34)^2\ (1.08)}\right]\right\}^{1/2}\right) = 1.29 \qquad [7.18]$$

and $W_{i2} = 1 - W_{i1}$, i.e. $W_{(ModelA2)2} = 1 - W_{(ModelA2)1} = 1 - (1.29) = (-0.29)$. The weighted sample means calculated using equation 7.12 is shown below. The weighted sample mean of the number of shortages, i.e. 4.56 is given in table 7.1.

$$\tilde{X}_{ModelA2}(21) = (1.29)(4.36) + (-0.29)(3.66) = 4.56 \qquad [7.19]$$

In this manner, the weighted sample means of all the three parameters of all the four models are calculated. The derived results are presented in the following sections.

7.5.1 Data Truncation

Data Truncation involves elimination of the initial and final states of output. Every simulation output is biased by the initial conditions of the model. It is crucial to define the initial conditions, as its effects are reflected on the model parameters. Therefore, by eliminating the data generated during its initial stages of run, one can minimize the errors for estimation of output parameters.

Every simulation experiment has the initial period of "warming up", after which the system attains, what is known as a "steady state" [Carrie, 1988]. It is recommended to use the output data for analysis only after the system has attained its "steady state". The initial "warming up" period varies from one system to another, and it is important to define the point at which the truncation needs to be made.

In this exercise, the initial conditions such as the initial tool stock level, in particular, has a significant impact on the number of tool shortages and average tool stock levels calculated over a period of time. Therefore, it became necessary to identify the system's approximate "warming up" time. It is usually, the point at which all the parts are loaded on the machines and when the machine shop is completely filled with the entities.

It was observed that the system took approximately two weeks of equivalent time to attain the steady state. The last part was being loaded in the week #7, which means the system would have started emptying before the beginning of week #7. Therefore, the data generated during the first two weeks and after the 6th week has been eliminated for analysis. The total period of four weeks was considered for generating the required output. These conditions apply to both the configurations.

7.6 RESULTS AND DISCUSSION

7.6.1 RESULTS :

As discussed in section 7.2, the simulation experiments were designed to obtain the data on three distinct parameters, viz; the 'Number of Purchases, the 'Number of Shortages' and the 'Average Tool Inventory Levels'. The aim was to observe the effect of changes in tool replenishment strategies on the production interruptions due to tool shortages.

As explained in the earlier section, the initial and final conditions of the simulation were eliminated for analysis. Therefore, the data used for analysis was extracted during the model's 'steady state' condition.

Each time the system registers a production stoppage, it records the details such as the job type, required tool type, total delay in processing that job which is regarded as 'one shortage'. Such shortages are registered in the 'shortage' database file.

Similarly, each time a tool is procured, it is recorded as one purchase in the 'purchase' database file. In A3, all the tools are bought on a regular basis (every week) irrespective of their inventory levels. In this case every tool type purchased is regarded as a single purchase.

The selected statistical method (as explained in section 7.5), was used to calculate the weighted sample means of all three performance measurement parameters in all the four models. The statistical method is made up of two stage sampling. In the first stage sampling, there were 20 replications taken for each model. Each replication was initiated by a unique negative seed value for generating random numbers (Foxpro-2 recommends use of negative seed values for maximum randomness, refer to section 7.6 for more details on random number generation). There were three runs (or number of observations, m=3) taken within each replication.

A second stage sampling was carried out using equation 7.10 for all the models. The weighted sample means for all three parameters were calculated using equations 7.10, 7.11 and 7.12. The final results for comparison are given in Tables 7.1, 7.2, 7.3 and 7.4.

Table 7.1 Comparison of Weighted Sample Means of *Purchases* and *Shortages* *(Configuration #1)*

Model Types	Number of Purchases	Number of Shortages
B1-(With TPS)	36	0
A1-(FIX-T,FLEX-P)	28.17	10.64
A2-(FLEX-T,FLEX-P)	34.93	4.56
A3-(FIX-T,FIX-P)	55	0.54

Table 7.2 Comparison of Inventory Levels- (Weighted Sample Means) *(Configuration #1)*

Tool Number	B1 (With TPS)	A1 (FIX-T, FLEX-P)	A2 (FLEX-T,FLEX-P)	A3 (FIX-T,FIX-P)
10	25.29	3.74	6	25.2
11	20.42	2.71	8	0.01
12	7	3.33	4	21
13	3.34	2.56	4	17.11
14	9	9	9	53
15	1	4	4	22
16	4.68	4	4	25
17	2.67	3.19	4.09	27
18	13.48	5.54	4.4	25.52
19	24.01	4.88	4.74	29.5
20	10	3	3	20.97

Table 7.3 Comparison of Weighted Sample Means of *Purchases* and *Shortages* *(Configuration #2)*

Model Types	Number of Purchases	Number of Shortages
B1-(With TPS)	43	0
A1-(FIX-T,FLEX-P)	36	19
A2-(FLEX-T,FLEX-P)	52	12
A3-(FIX-T,FIX-P)	80	14.75

Table 7.4 Comparison of Inventory Levels- (Weighted Sample Means) *(Configuration #2)*

Tool Number	B1 (With TPS)	A1 (FIX-T, FLEX-P)	A2 (FLEX-T,FLEX-P)	A3 (FIX-T,FIX-P)
10	24.75	4	6.25	18.25
11	30.25	3.25	8	2.5
12	10	3.5	4	17.25
13	3	3	4	13.5
14	34.25	8.75	9	38.75
15	1.25	4	4	17.5
16	4.25	3.75	4	20.25
17	9	4	4	17.25
18	20.75	2.5	4	14.25
19	24.25	4	5.25	24
20	15.5	1.5	3	2
21	3	5	5	16
22	0	6	6	30
23	28.25	1	3	1.25
24	14.75	4	4	15.75
25	7.5	3	3	16

7.6.2 DISCUSSION :

Ideally, a good tool planning system should incur minimum cost in purchases and face the least number of tool shortages. Table 7.1 and 7.3 shows that Model B1 (with TPS) has had no shortages during the entire period of simulation. Therefore, it can be concluded that the proposed TPS eliminates the possibility of tool shortages altogether.

In config #1 (Table 7.1), Model A3 (FIX-T, FLEX-P) shows minimal shortages. This model satisfies the requirements of zero shortages very closely (less than 1). However, A3 has the highest value of number of purchases and at least 150% times higher than B1. Furthermore, Table 7.2 and Fig 7.1 indicate that A3 has significantly higher inventory levels compared to all other models. This is also found to be true in configuration #2 (Table 7.3).

In configuration #1, the difference in the number of purchases between B1 (with TPS) and A2 (FLEX-T,FLEX-P) is very small, but A2 has at least four shortages during the same period. This means that at a very small extra cost of procurement in B1, one could eliminate the problem of tool shortages altogether. However, there are certain issues regarding the average inventory levels in these models. These are explained in the next section. Similar results are noticed in configuration #2 for Models B1 and A2. Both, the purchases and shortages are higher in A2 than B1.

In both the configurations, the number of purchases in A1 (FIX-T,FLEX-P) are 25% lower than B1 but with significantly higher number of tool shortages (19 in A1 as against zero in B1). The comparison of total cost savings between these two models can only be made if the unit shortage cost and unit purchase cost is known. However, it is anticipated that the implications of tool shortages on disruptions in production schedules would be greater and the indirect costs associated with such circumstances would be higher.

Comparison of inventory levels (for both the Configurations) :

In this exercise, the variety of tools is kept low (11 to 16 types), so that the volume of data for analysis could be reduced. The average inventory values of each of these tool types were recorded for all the models. Each configuration produced a set of results on inventory levels (Tables 7.2 and 7.4) which are presented graphically in Fig. 7.1 and 7.2 respectively. The weighted sample means of these tools were calculated by using the same method as explained in section 7.5. The following discussion applies to both the configurations, as the set of results obtained were similar.

Model-A3 (FIX-T,FLEX-P) shows the highest inventory level among all the models (except tool no.11). It means that it is the least cost effective model from the tool inventory perspective. Now, the real comparison lies between A1, A2 and B1.

It appears that A1 (FIX-T,FLEX-P) and A2 (FLEX-T,FLEX-P) have approximately similar level of inventory and one cannot confidently state which one of these offer higher savings. Therefore, it can be concluded that the difference in their tooling levels is minimal and insignificant compared to inventory levels in B1.

The inventory levels in B1 is much higher in certain tools compared to the same tools in A1 and A2 (e.g. tool no. 10, 11, 19, 20 and 23), See Graphs Fig 7.1 and 7.2. On the other hand, tool no. 15 in B1 indicates lower level than A1 and A2. The difference in inventory levels of all other types of tools in A1, A2 and B1 is minimal.

Further investigation into the tools with higher levels in B1, (no. 10, 11 and 19 in particular) was carried out. Two potential reasons were identified for such results, and they are,

(1) The tools in B1 are replenished at the beginning of each week and are gradually consumed during the later part of that week. In A1 and A2, as soon as the tools are restocked, they are issued to various workcenters, most of which have been waiting for these tools to arrive. Since, such tools do not spend much time in tool stores, their average inventory level is lower than B1.

(2) It was observed that the tools with higher levels in B1 necessarily have higher requirement throughout the simulation period. The shortages of tool no 10, 11 and 19 in A1 and A2, in both the configurations (Tables 7.5, 7.6, 7.7 and 7.8

respectively) substantiate the reason that their demand has been particularly higher than estimated. Increase in 'order size' or the 'minimum stock level' of these tools would reduce the shortages of these tools in A1 and A2. But this solution would have not been known without the help of the TPS. This is one of the advantages of the TPS that it can act as a firm guideline for estimating the requirements close to the realistic figures.

7.7 CONCLUSION

It can be summarised that the model with the Tool Planning System (TPS) (model-B1) shows the average expected inventory levels that achieves the goal of zero stoppages. By using the TPS, one could reduce the number of purchases and thus keep the cost of procurement to its minimum. It was observed that the changes in the configuration of the job shop model produced identical results. This means that TPS achieves the goal of zero stoppages with minimal inventory irrespective of the size of the production environment. Further indirect benefits of TPS include, a good foresight of expected periodic demand of tools, the level of tooling activities required within that period and supporting tool capacity planning activities. Such data can also be used for budgeting tools, and thus assisting the development of overall strategy for production.

The objective of establishing a structure of tool management with its primary functions was achieved. The data flow diagrams show the information needs of a typical TM system in a conventional manufacturing environment. This structure forms a suitable foundation to build any TM system.

The principles on which a tool planning system could be built were laid out. A generic methodology for planning tools in batch manufacturing was presented. The information model of a TPS was produced and verified using a database package.

The effectiveness of the TPS model was evaluated using simulation. The GASP methodology which acted as a simulation engine to run the experiments was employed and coded in a database package. This engine was successfully linked with the developed TPS Model. It appears that with the help of the TPS, the possibility of tool shortages within the batch manufacturing environment could be reduced. The accuracy of tooling requirements could be improved if the information such as tool life is estimated close to the realistic figures. The TPS model lacks facility for determining the requirements for multiple tools with complex assemblies. The concept of Bill of Tools (BOT) could be incorporated in this model to achieve this.

Table 7.5 Shortage Report of Model-A1 (FIX-T,FLEX-P) - Configuration #1

ORDER_NO	PART_NO	OPN_NO	TCODE	SHORT_QTY	WAIT_START	WAIT_END	TOTAL_WAIT
102	1000	10	11	8	73818	184474	110656
105	1000	10	11	5	73818	142978	69160
104	1003	30	11	6	96505	108000	11495
103	1002	30	10	2	155754	158000	2246
110	1003	20	10	2	161496	233985	72489
100	1000	40	19	3	173961	233985	60024
113	1003	20	10	4	193189	261967	68778
108	1003	30	11	4	204760	241480	36720
107	1002	30	10	5	204760	208000	3240
114	1003	20	10	3	291683	312608	20925
105	1000	40	19	4	291683	312608	20925
113	1003	30	11	4	298966	308932	9966
115	1002	30	10	6	327737	550000	222263
102	1000	40	19	5	362332	550000	187668
109	1000	40	19	6	381909	550000	168091
119	1002	30	10	4	406768	550000	143232

Table 7.6 Shortage Report of Model-A2 (FLEX-T,FLEX-P) - Configuration #1

ORDER_NO	PART_NO	OPN_NO	TCODE	SHORT_QTY	WAIT_START	WAIT_END	TOTAL_WAIT
102	1000	10	11	8	71052	139563	68511
105	1000	10	11	5	71052	225075	154023
100	1000	40	19	3	146604	158000	11396
113	1003	20	10	2	244570	259770	15200
117	1004	10	20	1	276794	289697	12903
102	1000	40	19	1	328835	550000	221165
105	1000	40	19	5	333815	340000	6185
109	1000	40	19	2	394230	416060	21830
119	1002	30	10	2	403196	406000	2804
115	1002	40	18	2	405342	428000	22658

Table 7.7 Shortage Report of Model-A1 (FIX-T,FLEX-P) - Configuration #2

ORDER_NO	PART_NO	OPN_NO	TCODE	SHORT_QTY	WAIT_START	WAIT_END	TOTAL_WAIT
126	1007	10	220000	1	50259	550000	499741
102	1000	10	211000	8	65266	169233	103967
105	1000	10	211000	5	65266	134179	68913
124	1005	20	223000	13	66957	550000	483043
129	1007	10	220000	5	91562	550000	458438
104	1003	30	211000	6	105865	108000	2135
127	1005	20	223000	9	112461	550000	437539
125	1006	30	223000	2	147041	163421	16380
128	1006	30	223000	2	147041	163421	16380
100	1000	40	219000	3	147041	158000	10959
130	1005	20	223000	14	156648	550000	393352
132	1007	10	220000	2	161817	550000	388183
123	1004	10	220000	4	176300	208000	31700
128	1006	40	211000	1	179645	208000	28355
125	1006	40	211000	3	188957	208000	19043
133	1006	10	218000	2	189163	232960	43797
134	1007	10	220000	4	198807	258000	59193
111	1002	30	210000	2	204380	208000	3620

108	1003	20	210000	2	242810	269330	26520
113	1003	20	210000	4	242810	269330	26520
103	1002	40	218000	2	242810	269330	26520
111	1002	40	218000	2	242810	269330	26520
101	1000	40	219000	4	242810	269330	26520
117	1004	10	220000	2	271222	363797	92575
107	1002	30	210000	3	278628	358000	79372
111	1002	50	220000	2	284930	308000	23070
114	1003	20	210000	5	284930	310576	25646
103	1002	50	220000	3	299130	308000	8870
115	1002	30	210000	3	311388	550000	238612
134	1007	40	223000	2	344106	389920	45814
118	1003	20	210000	4	345630	366515	20885
114	1003	30	211000	4	345630	418214	72584
102	1000	40	219000	6	383148	550000	166852
122	1003	20	210000	3	393892	550000	156108
121	1002	30	210000	3	461516	550000	88484
119	1002	30	210000	7	461516	550000	88484
118	1003	30	211000	4	461516	550000	88484
117	1004	40	218000	1	512930	550000	37070

Table 7.8 Shortage Report of Model-A2 (FLEX-T,FLEX-P) - Configuration #2

ORDER_NO	PART_NO	OPN_NO	TCODE	SHORT_QTY	WAIT_START	WAIT_END	TOTAL_WAIT
126	1007	10	220000	1	50492	550000	499508
102	1000	10	211000	8	64449	123439	58990
105	1000	10	211000	5	64449	202999	138550
124	1005	20	223000	13	74648	550000	475352
129	1007	10	220000	1	89240	550000	460760
127	1005	20	223000	9	122500	550000	427500
100	1000	40	219000	3	147155	158000	10845
132	1007	10	220000	2	162564	550000	387436
130	1005	20	223000	10	164125	550000	385875
110	1003	20	210000	3	245835	341260	95425
108	1003	20	210000	4	245835	341260	95425
113	1003	20	210000	4	245835	366386	120551
103	1002	40	218000	2	245835	341260	95425
102	1000	40	219000	5	369174	550000	180826
118	1003	20	210000	3	369174	378000	8826
117	1004	30	212000	2	369174	384824	15650
109	1000	40	219000	6	369174	384824	15650

115	1002	40	218000	3	369174	388624	19450
120	1001	40	218000	2	376585	404228	27643
107	1002	50	220000	1	407558	425086	17528
118	1003	30	211000	4	431906	471470	39564
122	1003	30	211000	5	431906	471470	39564
121	1002	30	210000	1	431906	471470	39564

Fig 7.1 COMPARISON OF INVENTORY LEVELS (Configuration #1)

Fig 7.2 COMPARISON OF INVENTORY LEVELS (Configuration #2)

DIRECTIONS FOR FUTURE RESEARCH

The developed TPS was tested in a 'push' type strategy, whereas many industries are adopting JIT philosophy. It is anticipated that the JIT strategy would have different effect on the tool planning methods. The TPS can be tuned to the 'pull' type of production system. Further work is recommended to testify the TPS under these manufacturing strategies.

The requirements generated by the TPS can also be used for costing purposes. An additional function on 'Tool Costing' can be included as a part of either the TM or the 'Costs and Accounts' function of the business. Further integration of this function is required with the TPS, to achieve better control over tool costs.

If the percentage of consumables versus returnables is changed, then it may have a significant impact on the tool inventory levels. Whether this has adverse or favourable effect can be determined by conducting further simulation studies. Advanced simulation models can be developed from the existing TPS to facilitate the above study. The results obtained can be used to recommend the percentage usage of returnables and disposables. It is anticipated that by using more consumables than returnables, the complexity of tool planning problems could be greatly reduced. Further work is required to support this hypothesis.

The developed TPS gives the tool plans in time buckets. But it is also important that these tools should be available at the required place on time and in good condition. A procedure for tracking the tools needs to be developed. This could determine the location of tools and estimate the time when the tool can be made available.

It is also equally important to have allocation methods in conjunction with the tool tracking procedures. The tools can be allocated to appropriate resources after tracking them. The survey indicated lack of suitable allocation methods in conventional manufacturing. Further work is recommended along these lines.

It seems that a method of tool life estimates is very important as the tool requirements quantity is based on this data. There are various factors that govern the tool life estimate, which include the workpiece material, cutting speed, feed and the material of the cutting tool. Since any tool is likely to be used more than once with different set ups, there is no criteria on which the estimates could be based. Further work is recommended to develop structured methods of estimating tool life.

In the simulation experiment, the Job Priority Rules are not defined (except the First-In-First-Out logic). It is anticipated that the TPS would consider the tool requirements for these jobs as this will be carried out before the jobs are released on to the shop floor. However, if the traditional stock control rules (as incorporated in type 'A' model), are used instead of the TPS, then it may have significant effect on the tool availability. A detailed study could aid in deducing either alternative stocking rules or changes in the job priority rules to ensure uninterrupted production.

The exercises carried out take into account only a single tool vendor. However, in practice, there are more than one vendors, in which case, the tool procurement time would vary. The suggested TPS model could be extended by incorporating this facility. Further work is required to study the effect of multiple vendors on tool availability.

It is also suggested that the viability of the tool planning methodology be evaluated outside the manufacturing sector, such as the Airline Industry. If the requirements of such service industries are different, then similar methodology applicable in these environments could be developed using some of the principles laid out in this work.

REFERENCES

Allock, A. : "Computer Suite Cuts Cell Software Costs", Machinery and Production Engineering, Vol 6/20 August, 1986.

Amoako-Gyampah, K., Meredith, J.R., Raturi, A. : "A comparison of tool management strategies and part selection rules for a flexible manufacturing system", International Journal of production research, vol 30, no 4, page 733-748, 1992.

Bao, H. : "Application of Dynamic programming to optimise tool replacement schedules for multi-tools operations", Transacations of ASME, Journal of Mechanical Design, Volume 102, July 1980.

Ber, A. and Falkenberg, D. : "A generic tool management system architecture", 3rd Biennial International Machine Tool Technical Conference, Session 6B, pp 87-109, Sept 3-10, 1986.

Broom, H.N. : "Production Management", Homewood, IL : Richard D. Irwin, Inc., 1967.

Carrie, Allan : 'Simulation of manufacturing systems', John Wiley and Sons, 1988.

Carrie, A.S., Perera, D.T. : "Work scheduling in FMS under tool availability constraints", International Journal of Production Research, 24, 1299-1308, 1986.

Carrie, A.S. and Bititci, U.S. : "Tool Management - A major challenge to integration.", The 23rd European Conference on Production and Inventory Control, Birmingham Nov 1988.

CCTA (Central Computer and Telecommunication Agency); Introduction to SSADM, 1990.

Chapman, B. : "Total Tool Management, The Big Puzzle", SME Technical Paper, Vol MS90-253, Page 2-8, 1990.

CTMS : Computerised Tool Management System, Amazon Computers Ltd., Sunrise Parkway, Linford Wood, Milton Keynes, MK14 6LQ, U.K.

Chakravarty, A.K., Shutb, A. : "Selecting parts and loading flexible manufacturing systems.", Proc. first ORSA/TIMS Conference on Flexible Manufacturing Systems, Ann Arbor, Michigan, 284-289, August 1984.

Cutts G.; Structured Systems Analysis and Design Methodology, Blackwell Scientific Publishing, 1987.

The first international conference on tool management and control, Grand Rapids

Chapter, APICS, Department of Management, Michigan State University, East Lansing, MI 48824-1121, Oct 12-13, 1988.

Galligan S.; Mokris M. : "Integrating tool control into a standard manufacturing system" Production and Inventory Management, First Quarter, 1981, pp 34-56.

Howe, D.R. : Data Analysis for Database Design, Edward Arnold Publications, 1983.

Hutchinson, G: "The impact of tooling on automated batch production", Proceedings International Machine Tool Design Conference, pp 445-452, September 1982.

ISTEL : "Storeman - Tool Management System", ISTEL Automation Ltd., Highfield House, Headless Cross Drive, Redditch, Worcestershire B97 5EG, U.K.

Kendall, L.A. and Sheikh, A.H. : "Tool replacement strategy for multi-tool machines using a probabilistic model", 4th North American Metalworking Research Conference, pp 416-419,1976.

Kravitt, D. : "Tool Management in the CIM environment", Tool Management and Control Conference, Michigan, pp 92-101, 1988.

Kuchinic, A.E, and Seidmann A. : "Tool Management in Automated Manufacturing - Operational Issues and Mathematical Models", Proceedings of International Industrial Engineering Conference, 1988.

LBMS (Learmonth and Burchett Management Systems) : LBMS-International HQ, Evelyn House, 62 Oxford Street, London W1N 9LF.

Liardet, M. and Whitehorn, M. : "Touching Base", Personal Computer Magazine, pp 186 - 213, Dec. 1991.

Lidbury, P. :"Airline Industry- Tool Management via Bar Coding", Epic Data Users Conference, Sept. 1989.

Long[1], M. : "Grass-roots tool management", American Machinist, page 52, May 1991.

Long[2], M. : "Before you install a tool management system", Tool Management - The new frontier for manufacturing cost control, Dec 8-10, 1991.

Maropoulos[1], P.J., Hinduja, S. : "Automatic Tool Selection for Finish Turning", Proceedings of Institution of Mechanical Engineers, Vol 204, pp 43-51.

Mason, F. : "The Future of Tool Management", Tool Management and control conference, pp 115-129, Michigan, 1988.

Mason, F. : "Tool System Pays for itself in the first year", pp 50, American Machinist, May 1991.

Melnyk, S. : "Tool Management and Control : An Introduction", Tool Management and control conference, Michigan, 1988.

Melnyk[1], S., Ghosh, S., Ragatz, G. : "Tooling constraints and shop floor scheduling : A simulation study", Journal of Operations Management, Vol.8, No.2, pp 69-89.

O'Grady, P.J., Menon, U.,: "Loading a flexible manufacturing system.", International Journal of Production Research, 25, 1053-1068, 1987.

Orlicky J : "Material Requirements Planning", McGraw Hill Book Company, 1975.

Perera, D.T.S. : 'The production planning problems of FMS with high tool variety, PhD Thesis, University of Strathclyde, Glasgow, 1988.

Perera, D.T.S. : "Tool Requirements Planning in FMS", Japan- USA Symposium on Flexible Automation, A Pacific Rim Conference, Vol III, pp 1119-1122, Kyoto Japan, July 1990.

Perera, D.T.S. and Carrie, A.S. :"Simulation of tool flow within a flexible manufacturing system", Proceedings of the 6th International Conference on FMs, pp 212-218, 1986.

Pritsekar, A.A.B. : 'The GASP-IV Simulation Language', John Wiley, New York, 1974.

Rajagopalan, S. : "Formulation and heuristic solutions for parts grouping and tool loading in flexible manufacturing systems.", Proc. of the second ORSA/TIMS Conference in Flexible Manufacturing Systems: Operations Research Models and Applications, Stecke,K.E. and R.Suri (eds), 311-320, 1986.

Ramlingam, S. ; Balsubramanian, N. and Peng Y. : "Tool life scatter, tooling cost and replacement schedules", 5th North American Metalworking Research Conference, pp 212-218,1976.

Ranky, P. : "A generic tool management system architecture for flexible manufactoring system", Robotica, volume 6, pp 221.234, 1988.

Rhodes, J.S. : "FMS Tool Management Systems", CASA/SME, MS86-169, Technical Paper, pp 1-22, 1986.

Sackett, P.J. and Cooper, D.J. : "Flexible manufacture through tool management in a high tool variety environment", proceedings of 3rd International Conference on FMS, Stuttgart, September 1984.

Samways, B. : Basic Facts - "Computers", Collins Gem Publications, 1989.

Sargent, R.G. : "Validation and Verification of Simulation Models", Proceedings of the 1992 Winter Simulation Conference, pp 104-114.

Stecke, K.E., : "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems", Management Science, 29, 273-288, 1983.

Stecke, K.E., Solberg, J.J. : "Loading and control policies for a flexible manufacturing system.", International Journal of Production Research, 19, 481-490, 1981.

TMS-2000 : "Tool Management System 2000", Cincinnati Milacron, Advanced Manufacturing Systems Division, 4701 Marburg Avenue, Cincinnati, Ohio 4701, U.S.A.

Traughber, T.J. : "Expert System for Tool Selection", Proceedings of Artificial Intelligence Conference (Paper from SAE International Congress and Exposition), pp 37-41, Feb 24-28, 1986.

Tyner, R.: "From Batch Manufacturing to Just-In-Time; its effect on the tool crib", Tool Management and control conference, pp 69-78, Michigan, 1988.

Veeramani, D.,Upton,D., Barash, M. : "Tool Management in Computer Integrated Manufacturing", Purdue University, School of Industrial Engineering Research Center for Intelligent Manufacturing Systems, West Lafayette, Indiana 47907.

Ventura, J.A. : "Grouping parts and tools in FMS production planning", International Journal of Production Research, Vol. 6, pp 1039, 1990.

Wadsworth, H.M. : "Handbook of Statistical Methods for Engineers and Scientists", McGraw Hill Publishing Company, 1989.

Wassweiler, W.R. : "Tool Requirements Planning", APICS Conference Proceddings, 1982.

Weimer, G.A. : "Automatic Selection Still Unfulfilled Promise", Iron Age, pp 98-105, Nov. 25, 1983.

Whitney, C.K., and Gaul, T.S. : "Sequential decision procedures for batching and balancing in FMSs.". Annals of Operations Research, 3, page 301-316, 1985.

Zhou, C. : "Tool Mnagement in Computer Integrated Manufacturing Systems", A PhD thesis in Industrial Engineering, The Graduate School, The Pennsylvania State University, USA, August 1988.

APPENDIX - 1

```
** The Program listing of Tool Planning System in Foxpro-2    *****
* MODELB-1 * Model with TPS
* tools either on time or delayed by 1 week or randomly delayed
* tool repl at fixed time intervals (50,000/weekly) -
* tools replenished until 350,000/week7, because last job released in week8
* tool inv recorded every 2000,
* time advance=2000,
* time finish=550,000
* 10 weekly plan
* random numbers for variables- plord_dates, batch size, operation times
* proposed variable is tool life, proc lead time etc

SET PROCEDURE TO c:\fox\vinay\gasp
DO environment
DO cleandata
DO variables              && setting the pointers
DO initialise             && data initialisation
DO random                 && random number generation
DO sim_variables
DO all_toolcal
DO week_req
DO firstjobs              && first job released(also start_op)
DO getatribs              && subsequent jobs released
DO engine
SET PROCEDURE TO C:\FOX\VINAY\PMENUB1
DO pmenub1
DO modmenu                && main menu procedure
*=======================================
PROCEDURE random
=RAND(-3)
FUNCTION irand
PARAMETER i, j
RETURN int((j-i+1)*rand()+i)
*=======================================
PROCEDURE cleandata
CLEAR                           &&clear the screen
CLOSE DATABASES
USE shortage
DELETE ALL
PACK
USE purchase
DELETE ALL
PACK
USE week
GO TOP
```

```
      REPLACE ALL WEEK1 WITH 0, WEEK2 WITH 0, WEEK3 WITH 0,
      WEEK4 WITH 0; WEEK5 WITH 0, WEEK6 WITH 0, WEEK7 WITH 0,
      WEEK8 WITH 0, WEEK9 WITH 0; WEEK10 WITH 0
SELECT F
USE tool_inv ORDER tcode
REPLACE ALL in_stock WITH 0, tot_stock WITH 0, ave_stock WITH 0;
  ordered WITH .N.
RETURN
*=========================================
PROCEDURE sim_variables
  PUBLIC nnmac,nstops,rel_time,ave_count
= RAND(-1)
  ave_count=0
  nstops=0
  nnmac=5
  PUBLIC machine[nnmac]
  j=1
  DO WHILE j<=nnmac
    machine[j]=0
    j=j+1
  ENDDO
RETURN
*=========================================
PROCEDURE all_toolcal           && agg tool req  for all parts
SELECT H
USE all_req ORDER ordop
DELETE ALL
PACK
SELECT G
USE part_ord ORDER plord_date
GO TOP
DO WHILE .NOT. EOF()
  mpartno=part_no
  m_size=batch_size
  m_orderno=order_no
  DO part_toolcal WITH mpartno,m_size
  SELECT A
  GO TOP
  DO WHILE .NOT. EOF()
    m_opnno=opn_no
    m_tcode=tcode
    m_wcentno=wcent_no
    SELECT H
    APPEND BLANK
    REPLACE order_no WITH m_orderno, opn_no WITH m_opnno;
```

```
    tcode WITH m_tcode, wcent_no WITH m_wcentno
    SELECT A
    SKIP 1
  ENDDO
  SELECT H
  GO TOP
  SET RELATION TO tcode INTO tool_req       && H - E
  SELECT E
  GO TOP
  SELECT H
  GO TOP
  SET FILTER TO order_no=m_orderno
  SCAN FOR tcode=E->tcode .AND. opn_no=E->opn_no
    REPLACE H->reqd_qty WITH E->reqd_qty;
    H->run_time WITH E->run_time
  ENDSCAN
  SET FILTER TO
  SELECT G
  SKIP 1
ENDDO
CLOSE DATABASES
RETURN
*========================================
PROCEDURE week_req            && weekly tool req for all parts in MPS
PRIVATE morderno,mreltime,mtcode,mrqty
SELECT B
USE tcode_na ORDER tcode
SELECT F
USE tool_inv ORDER tcode
SET RELATION TO tcode INTO tcode_na       && F - B
SELECT C
USE week ORDER tcode
SET RELATION TO tcode INTO tool_inv       && C - F
SELECT H
USE all_req ORDER ordop
SELECT G
USE part_ord ORDER order_no
GO TOP
DO WHILE .NOT. EOF()
  morderno=order_no
  mreltime=plord_date

* All jobs released within six weeks from the start of the simulation
* However, the simln period=10 weeks, the tool repl done for 6 weeks only
```

```
  DO CASE
    CASE mreltime>0 .AND. mreltime<50000
      x=1
    CASE mreltime>50000 .AND. mreltime<100000
      x=2
    CASE mreltime>100000 .AND. mreltime<150000
      x=3
    CASE mreltime>150000 .AND. mreltime<200000
      x=4
    CASE mreltime>200000 .AND. mreltime<250000
      x=5
    CASE mreltime>250000 .AND. mreltime<300000
      x=6
    CASE mreltime>300000 .AND. mreltime<350000
      x=7
    CASE mreltime>350000 .AND. mreltime<400000
      x=8
    CASE mreltime>400000 .AND. mreltime<450000
      x=9
    CASE mreltime>450000 .AND. mreltime<500000
      x=10

  ENDCASE

  field='week'+ALLTRIM(STR(x))

  SELECT H
  DO WHILE order_no=morderno
    mtcode=tcode
    mrqty=reqd_qty
    SELECT C
    SEEK mtcode
    IF FOUND()
      REPLACE &field WITH (&field+mrqty)
    ENDIF
    SELECT H
    SKIP 1
  ENDDO

  SELECT G
  SKIP 1
ENDDO
RETURN
```

```
*=========================================
PROCEDURE part_toolcal  && tool reqments for each part/order
PARAMETERS part_no,m_batch_size
* 'WHAT' and 'HOW MUCH OF EACH'
SELECT B
USE tcode_na ORDER tcode
part=STR(part_no,4,0)
fname='p'+part+'.dbf'
SELECT A
USE &fname ORDER tcode
SET RELATION TO tcode INTO tcode_na         && A - B

* setting up the relationships between  A - B - D - E

SELECT E
USE tool_req ORDER tcode
DELETE ALL
PACK
SELECT D
USE tool_lif ORDER tcodeid
SELECT A
SET RELATION TO tcode INTO tool_req         && A - E
SELECT B
SET RELATION TO tcode INTO tool_req         && B - E
SELECT D
SET RELATION TO tcode INTO tool_req         && E - D
*
* data processing begins here *
*
SELECT A                   && process plan database
GO TOP
DO WHILE .NOT.EOF()
  m_opnno=opn_no
  m_tcode=tcode
  m_runtime=run_time
  SELECT E
  APPEND BLANK
  REPLACE opn_no WITH m_opnno, tcode WITH m_tcode;
  run_time WITH m_runtime
  SELECT A
  SKIP 1
ENDDO
SELECT E
GO TOP
```

```
DO WHILE .NOT.EOF()
  mmtcode=tcode
  SELECT B
  GO TOP
  SEEK mmtcode
  IF FOUND()
    IF returnable=.T.
      total_jobs=INT(D->unit_life/E->run_time)
      no_tchanges=INT(m_batch_size/total_jobs)
      no_regrinds=INT(D->est_life/D->unit_life)
      scrap_qty=INT(no_tchanges/no_regrinds)
      qty_ret=INT(scrap_qty)+1
      life_left=(qty_ret*D->est_life)-(no_tchanges*D->unit_life)
      IF life_left=0                &&such values for individual orders
        qty_ret=qty_ret-1           &&are not taken into account
      ENDIF                         &&more than one order will have more than one
      ltcode=mmtcode
      SELECT D
      GO TOP
      LOCATE FOR D->tcode=ltcode
      IF FOUND()
        REPLACE D->ava_life WITH life_left     &&FOR D->tcode=mmtcode
*         SET PRINTER ON
*         ?'tcode' AT 2,D->tcode AT 10,'ava_life'AT 20,D->ava_life AT 30
*         SET PRINTER OFF
      ELSE
        @20,50 SAY "TOOL NOT FOUND"
        WAIT
      ENDIF
      REPLACE E->reqd_qty WITH qty_ret       && note its important
    ELSE
      qty_con=INT((m_batch_size * A->run_time)/(D->est_life)) + 1
      REPLACE E->reqd_qty WITH qty_con
    ENDIF
  ELSE
    DO error WITH PROGRAM(),LINENO()
  ENDIF

  SELECT E
  SKIP 1
ENDDO
RETURN
*========================================
PROCEDURE firstjobs      && introducing first job in for sim
* dummy entity for advancing tnow by 2000 * eve code 100
```

```
p=1
DO WHILE p<=nnatr
 atrib[p]=0
 p=p+1
ENDDO
atrib[2]=100             && eve 100 for dummy entity
DO queue WITH 1          && delivering it to event queue
RETURN
*=====================================================
PROCEDURE event
PARAMETERS jevent
?tnow AT 10
 DO CASE
  CASE jevent=1
   DO end_op

  CASE jevent=100                && for both time advance and
   DO CASE                       && weekly replenishments**
     CASE atrib[1]=0
      DO weekly_repl WITH 1      && restocking for week 1
    CASE atrib[1]=50000
      DO weekly_repl WITH 2
    CASE atrib[1]=100000
      DO weekly_repl WITH 3
    CASE atrib[1]=150000
      DO weekly_repl WITH 4
     CASE atrib[1]=200000
      DO weekly_repl WITH 5
     CASE atrib[1]=250000
      DO weekly_repl WITH 6
     CASE atrib[1]=300000
      DO weekly_repl WITH 7
     CASE atrib[1]=350000
      DO weekly_repl WITH 8
   ENDCASE

   atrib[1]=tnow+2000
   DO queue WITH 1
   IF tnow>=100000.AND.tnow<=300000
*
   SELECT F                      && for average stock levels
   USE tool_inv ORDER tcode
   GO TOP
   DO WHILE .NOT.EOF()
    m.instock=in_stock
```

```
        m.tstock=tot_stock
        sum=(m.instock+m.tstock)
        REPLACE tot_stock WITH sum
        SKIP 1
      ENDDO
      ave_count=ave_count+1
*
      ENDIF
          DO waitq

    CASE jevent=200
          arrq=atrib[6]*2
          DO queue WITH arrq
          DO waitq

  ENDCASE
RETURN
*=========================================
PROCEDURE weekly_repl
PARAMETERS y
field='week'+ALLTRIM(STR(y))
SELECT J
USE purchase
SELECT C
USE week ORDER tcode
SET RELATION TO tcode INTO tool_inv                 && C - F
SELECT F
USE tool_inv ORDER tcode
GO TOP
DO WHILE .NOT. EOF()
  m_tcode=tcode
  oldstock=in_stock
  SELECT C
  SEEK m_tcode
  IF FOUND()
    newstock=C->&field
    IF newstock>0
      sum1=oldstock+newstock
      SELECT F
      REPLACE F->in_stock WITH sum1
      IF tnow>=100000.AND.tnow<=300000
        SELECT J                   &&recording purchase details
        APPEND BLANK
        REPLACE tcode WITH m_tcode, tordered WITH tnow;
          qty_ord WITH newstock
```

```
    ENDIF
   ENDIF
  ELSE
   DO error WITH PROGRAM(),LINENO()
  ENDIF
  SELECT F
  SKIP 1
ENDDO
RETURN
*=========================================
PROCEDURE getatribs
* insert attribute values for jobs from MPS and putting them
* in nset and queuing in event queue for release.
* jobs in MPS already indexed on plord_date in increasing order

SELECT G                  && part_ord/(MPS)
USE part_ord ORDER plord_date
GO TOP
z=200                     && initial setting of event code =200
DO WHILE .NOT.EOF()       && z is the event code variable
  mrel_date=plord_date    && for increaments of 200 for each job
  x=mrel_date+2000
  y=mrel_date-2000
  m.plord_date=irand(x,y)
  atrib[1]=m.plord_date
  atrib[2]=z
  atrib[3]=part_no
  msize=batch_size
  p=msize*(1.05)          && about +-5% variation in batch size
  q=msize*(0.95)
  bsize=irand(p,q)        && random variation of batch size
  atrib[4]=bsize
  m.order_no=order_no
  atrib[10]=order_no
  part=STR(part_no,4,0)
  fname='p'+part+'.dbf'    && P1000.dbf a process plan
  SELECT A
  USE &fname ORDER opn_no
  GO TOP
  atrib[5]=opn_no
  atrib[8]=tcode
  m.opnno=opn_no
  m.tcode=tcode
  SELECT H
  USE all_req ORDER ordop
```

```
  GO TOP
  SET FILTER TO order_no=m.order_no
  LOCATE FOR tcode=m.tcode .AND. opn_no=m.opnno
  IF FOUND()
    atrib[6]=wcent_no
    opn_time=run_time
    v=opn_time*(1.15)        && +-15% variation in opn time
    u=opn_time*(0.85)
    runtime=irand(v,u)
    atrib[7]=runtime
    atrib[9]=reqd_qty
  ELSE
    DO error WITH PROGRAM(),LINENO()
  ENDIF
  SET FILTER TO
  DO queue WITH 1          && delivering it to event calender
  SELECT G
  SKIP 1
ENDDO
RETURN
*=========================================
PROCEDURE waitq
PRIVATE toolcode,tqty
jmac=1
DO WHILE jmac <= nnmac          && doing for all machines 1 by 1
  IF machine[jmac]=0            && and if the machine is idle
    stopq=(jmac*2)+1            && defining stopq nos.
    njobs=nnq[stopq]            && total entities/jobs in stopq
    waitjob = mfe(stopq)        && mem loc of first ent. in stopq
    IF njobs>0                  && if there are jobs in stopq
      DO WHILE njobs>0          && do until there are jobs in stopq
        pv=1
        DO WHILE pv<=nnatr
         atrib[pv]=nset[waitjob+pv]
         pv=pv+1
        ENDDO
        toolcode=atrib[8]
        tqty=atrib[9]
        order=atrib[10]
        SELECT F
        USE tool_inv ORDER tcode
        GO TOP
        SEEK toolcode                 && check for tool availability
        IF FOUND()
          IF (F->in_stock - tqty) >= 0    && if no shortages
```

```
*        ?"wait queue information WHEN NO SHORTAGES" AT 2
*        DO atribdis
        SELECT B
        USE tcode_na ORDER tcode
        LOCATE FOR tcode=toolcode
        IF FOUND()
          IF returnable=.T.            && for returnables return -
            SELECT D                   && one extra to the stores
            USE tool_lif ORDER tcodeid
            IF D->ava_life<>0          && if life left as per -
              tqty=tqty-1              && tool cals is 0 then throw away
            ENDIF
          ENDIF
        ENDIF

        SELECT J         && record the shortage details
        USE shortage
        GO TOP
        DO WHILE .NOT.EOF()
                                                                    I     F
           J->order_no=atrib[10].AND.J->part_no=atrib[3].AND.J->opn_no
            =atrib[5]J->tcode=atrib[8]
            REPLACE wait_end WITH tnow
            EXIT
          ENDIF
          SKIP 1
        ENDDO

        REPLACE F->in_stock WITH (F->in_stock - tqty)
        DO remove WITH waitjob,stopq
        total_time=ATRIB[4]*ATRIB[7]       && cal total opn time
        ATRIB[1]=tnow+total_time      && advance sim time
        ATRIB[2]=1                    &&set eve code to 1
        machine[jmac]=1               && set machine 'busy'
        DO queue WITH 1               && deliver to eve queue
        njobs=njobs-1                 && set do loop for exit
        waitjob=mfe[stopq]
       ELSE                           && if shortages
        mdiff=(F->in_stock - tqty)*(-1)
        njobs=njobs-1
        waitjob=nset[waitjob+nnatr+1]
        v=1
        DO WHILE v<=nnatr
          atrib[v]=nset[waitjob+v]
          v=v+1
```

```
        ENDDO
      ENDIF
    ELSE
      DO error WITH PROGRAM(),LINENO()
    ENDIF
   ENDDO
  ENDIF
 ENDIF
 DO start_op WITH jmac                && then go to routine for arrq
 jmac=jmac+1                          && next machine
ENDDO
RETURN
*=========================================
PROCEDURE start_op
PARAMETERS jmac
PRIVATE mtcode,rqty
 IF machine[jmac]=0
  arrq=jmac*2
  stopq=arrq+1
  nextjob = mfe(arrq)
  ncount=nnq[arrq]
  IF ncount>0
   DO WHILE ncount>0
    v=1
    DO WHILE v<=nnatr
     atrib[v]=nset[nextjob+v]
     v=v+1
    ENDDO
    mtcode= atrib[8]
    rqty=atrib[9]
    SELECT F
    USE tool_inv ORDER tcode
    GO TOP
    SEEK mtcode
    IF FOUND()
     IF (F->in_stock - rqty) < 0          && if tool shortages
      mdiff=(F->in_stock - rqty)*(-1)
      REPLACE F->in_stock WITH 0          &&testing for cumulative diff
      DO remove WITH nextjob,arrq
      SELECT J
      USE shortage
      APPEND BLANK
      REPLACE order_no WITH atrib[10];
       opn_no WITH atrib[5], tcode WITH atrib[8];
       short_qty WITH mdiff, wait_start WITH tnow;
```

```
          part_no WITH atrib[3]
         DO queue WITH stopq
         ncount=ncount-1
         nextjob=mfe[arrq]
        ELSE                              && if no shortages
         SELECT B
         USE tcode_na ORDER tcode
         LOCATE FOR tcode=mtcode
         IF FOUND()
          IF returnable=.T.               && for returnables return
           SELECT D                       && one extra to the stores
           USE tool_lif ORDER tcodeid
           IF D->ava_life<>0              && if all life consumed
            rqty=rqty-1                   && then throw away
           ENDIF
          ENDIF
         ENDIF
         REPLACE F->in_stock WITH (F->in_stock - rqty)
         DO remove WITH nextjob,arrq
         total_time=ATRIB[4]*ATRIB[7]
         ATRIB[1]=tnow+total_time
         ATRIB[2]=1
         machine[jmac]=1
         DO queue WITH 1
         ncount=ncount-1
         nextjob=mfe[arrq]
        ENDIF
       ELSE
        ncount=ncount-1
        nextjob=mfe[arrq]
        DO error WITH PROGRAM(),LINENO()
       ENDIF
      ENDDO
     ENDIF
    ENDIF
RETURN
*===================================================
PROCEDURE end_op
 m.wcent_no=atrib[6]
 machine[m.wcent_no]=0
 m.next_op=atrib[5]+10
 m.part_no=atrib[3]
 m.order_no=atrib[10]
 SELECT H
 GO TOP
```

```
  SET FILTER TO order_no=m.order_no
  GO TOP
  LOCATE for opn_no=m.next_op
  IF FOUND()
    atrib[5]=opn_no
    atrib[6]=wcent_no
    atrib[7]=run_time
    atrib[8]=tcode
    atrib[9]=reqd_qty
    arrq=2*atrib[6]
    DO queue WITH arrq
  ENDIF
  SET FILTER TO
  DO waitq
RETURN
*=====================================
```

APPENDIX - 2

```
* The Program listing of GASP Methodology in Foxpro-2 **************
*GASP.PROGRAM - The simulation engine
*LIBRARY OF PROCEDURES
PROCEDURE variables      && Declare global variables.
PUBLIC mfa,nnapt,nnapo,nnfil,nnatr,nntry
PUBLIC ttnex,ttfin,tnow
PUBLIC firstjob
mfa=1
nnfil=20
nnatr=10
nntry=100
nnapo=nnatr+1
nnapt=nnatr+2
nsize=nnapt*nntry
tnow=0
ttfin=500000
PUBLIC nnq[nnfil],mfe[nnfil], mle[nnfil],kknk[nnfil],iinn[nnfil]
PUBLIC nset[nsize],atrib[nnatr],atr[nnatr]
RETURN
*****************************************************************
PROCEDURE initialise
*
*initialize NSET() array pointers
i=1
DO WHILE i <= nntry
  icsuc=i*nnapt
  icprd=icsuc-nnapo
  nset[icprd]=-1
  nset[icsuc]=icsuc+1
  i=i+1
ENDDO
nset[icsuc]=0
mfa=1
*
j=1
DO WHILE j<=nnfil
  nnq[j]=0
  mfe[j]=0
  mle[j]=0
  iinn[j]=1
  kknk[j]=2
  j=j+1
ENDDO
iinn[1]=1
RETURN
```

```
*****************************************************
PROCEDURE engine        && Simulation Executive
DO WHILE tnow < ttfin
  IF ttnex > ttfin
    ttnex=ttfin
  ENDIF
  IF ttnex < tnow
    DO error WITH PROGRAM(),LINENO()
  ENDIF
  nent=nnq[1]
  DO CASE
    CASE nent<0
      DO error WITH PROGRAM(),LINENO()
    CASE nent=0
      DO error WITH PROGRAM(),LINENO()
    OTHERWISE
      nexte=mfe[1]
      tnow=nset[nexte+1]
      jevent=nset[nexte+2]
      IF tnow<ttfin
        DO remove WITH nexte,1
        DO event WITH jevent
      ENDIF
  ENDCASE
ENDDO
RETURN
*****************************************************************
PROCEDURE remove
PARAMETERS ke,kq
*Removes the 'ke'th entry from the 'kq'th queue
je=ke
jq=kq
*
ja=1
DO WHILE ja<=nnatr
  nsisa=ke+ja
  atrib[ja]=nset[nsisa]
  ja=ja+1
ENDDO
*
*updating the pointer
nsisa=nnapo+je
jk=nset[je]
jl=nset[nsisa]
nset[nsisa]=mfa
```

```
mfa=je
nsisa=jk+nnapo
IF jl<=0
  IF jk<=0
    mfe[jq]=0
    mle[jq]=0
  ELSE
    nset[nsisa]=0
    mle[jq]=jk
  ENDIF
ELSE
  IF jk<=0
    nset[jl]=0
    mfe[jq]=jl
  ELSE
    nset[nsisa]=jl
    nset[jl]=jk
  ENDIF
 ENDIF
*updating the next event time
IF jq=1
  nexte=mfe[1]
  IF nexte<0
    DO error WITH PROGRAM(),LINENO()
  ELSE
    IF nexte=0
      ttnex=ttfin
    ELSE
      ttnex=nset[nexte+1]
      IF ttnex>ttfin
        ttnex=ttfin
      ENDIF
    ENDIF
  ENDIF
ENDIF
*update no of entries in the queue
nnq[jq]=nnq[jq]-1
RETURN
******************************************************************
PROCEDURE queue
PARAMETERS kq
*queue the entities(specified in the atrib[] array into queue 'kq'
*
*check the queue no
IF kq<1
```

```
  DO error WITH PROGRAM(),LINENO()
ENDIF
*
*check the no of entities in the queue
nent=nnq[kq]
IF nent<0
  DO error WITH PROGRAM(),LINENO()
ENDIF
*
*check whether the space is available for entity being used
IF mfa<0
  DO error WITH PROGRAM(),LINENO()
ENDIF
*
*store attributes of the entity(record no=mfa)
i=1
DO WHILE i<=nnatr
  nsisa=mfa+i
  nset[nsisa]=atrib[i]
  i=i+1
ENDDO
*
*set the next available record no (=mfa)
new=mfa
mfa=nset[nsisa+1]
*If there are no entities in the queue
IF nent=0
  nset[new]=0
  nset[new+nnapo]=0
  mfe[kq]=new
  mle[kq]=new
ENDIF
* If there is at least one entity in the queue.
IF nent>0
  mfex=mfe[kq]
  mlex=mle[kq]
  IF kq=1
    ks=1
    inns=1
    qqind=1
  ELSE
    ks=kknk[kq]
    inns=iinn[kq]
  ENDIF
  DO CASE
```

```
    CASE inns <= 2
     qind=3-2*inns
     nsisa=new+ks
     nsisb=mlex+ks
     diff=(nset[nsisa]-nset[nsisb])*qind
     DO WHILE diff<0 .AND. mlex>0
            mlex=nset[mlex]
            IF mlex>0
              nsisb=mlex+ks
              diff=(nset[nsisa]-nset[nsisb])*qind
           ENDIF
     ENDDO
*
     IF diff=0 .AND. kq=1
       ns1=new+1
       ns2=mlex+1
       diff1=ns1-ns2
       nsisa=new+2
            nsisb=mlex+2
       diff=nset[nsisa]-nset[nsisb]
            DO WHILE diff<0 .AND. mlex>0 .AND. diff1=0
              mlex=nset[mlex]
         IF mlex>0
               nsisb=mlex+2
               diff=nset[nsisa]-nset[nsisb]
          ENDIF
       ENDDO
     ENDIF
*
     IF mlex<=0
       msu=mfe[kq]
        nset[new]=0
            nset[new+nnapo]=msu
       nset[msu]=new
       mfe[kq]=new
     ELSE
            msu=nset[mlex+nnapo]
            IF msu=0
              nset[new]=mlex
              nset[mlex+nnapo]=new
              nset[new+nnapo]=0
              mle[kq]=new
            ELSE
              nset[mlex+nnapo]=new
              nset[new]=mlex
```

```
            nset[new+nnapo]=msu
            nset[msu]=new
          ENDIF
    ENDIF
*
   CASE inns=3
    mlex=mle[kq]
    nset[mlex+nnapo]=new
    nset[new]=mlex
    nset[new+nnapo]=0
    mle[kq]=new
*
   CASE inns=4
    mfex=mfe[kq]
    nset[mfex]=new
    nset[new]=0
    nset[new+nnapo]=mfex
    mfe[kq]=new
  ENDCASE
ENDIF
nnq[kq]=nnq[kq]+1
*then update next event time
IF kq=1
  nexte=mfe[1]
  ttnex=nset[nexte+1]
ENDIF
*DO qdata
*DO disnset
RETURN
*********************************************************
PROCEDURE error
PARAMETERS x_program,x_lineno
? 'ERROR: ', ' PROGRAM: ', x_program, ' LINENO: ',x_lineno
WAIT
CANCEL
RETURN
*********************************************************
PROCEDURE environment
SET TALK off
SET ECHO off
SET DATE BRITISH
SET CONFIRM ON
RETURN
*********************************************************
```

```
PROCEDURE disnset
i=1
j=1
DO WHILE i <= nntry
  k=0
  DO WHILE k< nnapt
    ?? NSET(j+k),' '
    k=k+1
  ENDDO
  ?''
  j=j+nnapt
  i=i+1
ENDDO
RETURN
*************************************************************
PROCEDURE qdata
*SET PRINTER on
* Produce detail queue information
q=1
DO WHILE q<= nnfil
  IF nnq(q)>0
  ?'******',tnow, '*********'
  ?' '
  ?' '
  ? 'QUEUE NO: ' AT 1, q AT 11
  ? '========== ' AT 1
  ?
  ? 'NO. OF ENTITIES :' AT 3, nnq(q) FUNCTION '999' AT 20
  ? 'FIRST ENTRY :' AT 3, mfe(q) FUNCTION '999' AT 20
  ? 'LAST ENTRY : ' AT 3, mle(q) FUNCTION '999' AT 20
  ?
  e=1
  no_entities = nnq(q)
  first = mfe(q)
  DO WHILE e <= no_entities
    ?
    ?
    ?'Entity:',e
    a=1
    DO WHILE a <= nnatr
      ?a, nset(first+a)
      a=a+1
    ENDDO
    x=first+nnapo
    first=nset(x)
```

```
   e=e+1
*   WAIT
  ENDDO
  ENDIF
  q=q+1
ENDDO
*SET PRINTER off
RETURN
***************************************************
```

APPENDIX - 3

Running the simulation models in FOXPRO-2

The developed models in this project can be used to learn the concepts of the proposed Tool Planning System and compare its performance with traditional stock control approaches. The implications of various input parameters such as Production Plans, Tool details and Process Plans on the performance measurement parameters like Shortages, Purchases and Inventory levels can be studied with the help of these models.

The floppy disc (3.5 inch.) which contains all the files required to run the simulation, is attached with this thesis. The user is advised to use the Personal Computer with microprocessor 80286 and upwards with at least 4mb of RAM. It is essential that FOXPRO-2 should be installed on the hard disc of the PC before running the simulation. The user is advised to learn the basic command of Foxpro. It is recommended that all the files from the floppy disc to be copied in a separate directory. Let's call this "*sim*".

The program files of the four models are deginated as follows,

(1) 'Modela1.prg' (FIX-T,FLEX-P)

(2) 'Modela2.prg' (FLEX-T,FLEX-P)

(3) 'Modela3.prg' (FIX-T,FIXP-P)

(4) 'Modelb1.prg' (With TPS)

and the simulation is engine designated by 'Gasp.prg' (program listings is available in Appendix-2).

It is important to set the path to access these files from Foxpro by using the command in Foxpro Command Window, "SET DEFAULT TO [*drive c or d]*:*sim*".

The user is allowed to change the input data on Production Plan, the Process Plan, Tool Life details, initial inventory levels and Tool Master database. This can be achieved by using the required database files (as listed in chapter 6) and study its effect on the shortages, inventory levels and purchases. The program supplied simulates the job shop model with configuration #2, defined in Chapter 6.

The database files can be used by command "USE [*filename.dbf*]" and then could be modified by using "BROWSE" command or "APPEND" for adding more records. The database filenames and its details can be obtained from either opening these files in foxpro or from Chapter 6.

Once all the input data files are ammended, then the main menu for ruuning simulation models can be activated by the command, "DO modmenu". Appropriate selection of model can be made using the 'cursor' keys and then 'return' key. The program begins to run with the execution of procedures as described in Figs 7.3 and 7.4 in Chapter 7.

The numbers appearing on the screen represent the simulation clock in time units. At each event execution, the clock time is displayed. This indicates how long the simulation has been running for and when it will terminate. The terminating time is 550,000 time units which is the end of the eleventh week.

The user is given the choice of printing simulation reports at the end of each run. The main menu of model selection is activated after these print options. "ESC" key could be used, if the user wishes to quit the main menu and use the command window instead.

The TPS output is stored in 'Week.dbf', whereas, the simulation ouput is stored in three different databases, viz;

(1) Purchase.dbf

(2) Shortage.dbf and

(3) Tool_inv.dbf

These can be viewed by using either "BROWSE" OR "LIST" command. The typical shortage reports are given in Tables 8.3 and 8.4 in Chapter 8. The Foxpro session can be terminated by using the command "QUIT".